NAMES: Abdul Mueed, Abdullah Ch

ROLL NUMBERS: BSCS 18056, BSCS18055

D-S Assignment

PART a)

#include <iostream>

using namespace std;

#include <ctime>

#include <cmath>

//QUESTION NO 1

//PART a)

// .

bool IsPrime(long long Number) {

if (Number == 2)

{

return true;

}

else if (Number<2)

{

return false;

}

long long Limit = Number / 2;

for (int counter = 2; counter <= Limit; counter++)

{

if (Number % counter == 0) {

return false;

}

}

return true;

}

// .

bool IsPrimeFaster(long long Number)

{

if (Number == 2)

{

return true;

}

else if (Number<2)

{

return false;

}

long long Limit = sqrt(Number);

for (int counter = 2; counter <= Limit; counter++)

{

if (Number % counter == 0)

{

return false;

}

}

return true;

}

//.

// EXECUTION TIME

//The execution time for the number 2147483647 is 27.00 seconds by n/2 method

//and the execution time for the number 2147483647 is 0.00 seconds by sqrt(n) method

//PART b)

bool P(long long n)

{

long long N = (n\*n) + n + 41;

if (IsPrimeFaster(N) == true)

{

return true;

}

else if (IsPrimeFaster(N) == false)

{

return false;

}

}

//PART c)

void twinprime(long long range)

{

for (long long start = range; start > 0; start--)

{

if (IsPrimeFaster(start) == true && IsPrimeFaster(start - 2)==true)

{

cout << "Twin Primes are: " << start << "," << start - 2 << endl;

break;

}

}

}

//Part c

//Problem 2

bool EularConjucture(long long a, long long b, long long c, long long d)

{

if (pow(a, 4) + pow(b, 4) + pow(c, 4) == pow(d, 4))

{

return true;

}

else if (pow(a, 4) + pow(b, 4) + pow(c, 4) != pow(d, 4))

{

return false;

}

}

//problem 3

//part a

bool ellipticcurve(long long a, long long b, long long c)

{

if (313 \* (pow(a, 3) + pow(b, 3)) == pow(c, 3))

{

return true;

}

else if (313 \* (pow(a, 3) + pow(b, 3)) != pow(c, 3))

{

return false;

}

}

//part a

//part b

void goldbach(long long even)

{

long long half = even / 2;

if (IsPrimeFaster(half) == true)

{

cout << "The even number with the sum of two prime numbers is: " << half << "+" << half << endl;

}

else if (IsPrimeFaster(half) == false)

{

long long p1, p2 = 0;

long long countx = 0, county = 0;

for (long long x = half, y = half; x > 0, y < INT\_MAX; x--, y++, countx++, county++)

{

if (IsPrimeFaster(x) == true && IsPrimeFaster(y) == true && countx == county)

{

p1 = x;

p2 = y;

break;

}

}

cout << "The even number with the sum of two prime numbers is: " << p1 << "+" << p2 << endl;

}

}

//part b

//part c

bool IsEven(long long Number) {

if (Number % 2 == 0)

{

return true;

}

return false;

}

int Collatz(long long Number) {

if (Number / 2 == 1)

{

cout << "The number that is shrunked by dividing it by 2 is " << Number << endl;

cout << "It indeed shrunk to 1" << endl;

return 0;

}

else if (IsEven(Number) == true)

{

cout << "The number that is shrunked by dividing it by 2 is " << Number << endl;

long long Temp\_for\_Even = Number / 2;

cout << "The number shrunked is " << Temp\_for\_Even << endl;

return (Collatz(Temp\_for\_Even));

}

else if (IsEven(Number) == false)

{

cout << "The number that is tripled by multiplying it by 3 and adding 1 is " << Number << endl;

long long Temp\_for\_ODD = (3 \* Number) + 1;

cout << "The number increased is" << Temp\_for\_ODD;

return (Collatz(Temp\_for\_ODD));

}

}

//part c

//problem 3

int main() {

cout << "Enter the number of the problem to check that problem (1 to 3)" << endl;

int problem;

cin >> problem;

system("cls");

//problem 1

if (problem == 1)

{

cout << "Enter the part of the question you want to check. Enter a,b or c to check." << endl;

char part;

cin >> part;

system("cls");

//part a

if (part == 'a')

{

cout << "Enter n if you want to check by n/2 method otherwise enter s if you want to check by square root method: " << endl;

char check;

cin >> check;

if (check == 'n')

{

system("cls");

cout << "Enter the number you want to check" << endl;

long long number = 0;

cin >> number;

time\_t begin, end;

time(&begin);

system("cls");

if (IsPrime(number) == true)

{

cout << "its a prime number" << endl;

}

else if (IsPrime(number) == false)

{

cout << "its not a prime number" << endl;

}

time(&end);

double difference = difftime(end, begin);

printf("time taken by function to get executed function() %.2lf seconds.\n", difference);

}

else if (check == 's')

{

system("cls");

cout << "Enter the number you want to check" << endl;

long long number = 0;

cin >> number;

time\_t begin, end;

time(&begin);

system("cls");

if (IsPrimeFaster(number) == true)

{

cout << "its a prime number" << endl;

}

else if (IsPrimeFaster(number) == false)

{

cout << "its not a prime number" << endl;

}

time(&end);

double difference = difftime(end, begin);

printf("time taken by function to get executed function() %.2lf seconds.\n", difference);

}

//part a

}

//part b

else if (part == 'b')

{

cout << "Press u to enter user defined values to check if the proposition p(n)=n^2+n+41 is prime or not else press c to check at which number the proposition fails:" << endl;

char choice;

cin >> choice;

system("cls");

if (choice == 'u'){

cout << "Enter the number n to check p(n)=n^2+n+41 is prime or not" << endl;

long long n;

cin >> n;

if (P(n) == true)

{

cout << "its a prime number" << endl;

}

else if (P(n) == false)

{

cout << "its not a prime number" << endl;

}

}

else if (choice == 'c')

{

for (int n = 0; n < INT\_MAX; n++)

{

if (P(n) == false)

{

cout << "Claim failed at the number:" << n << endl;

break;

}

}

}

}

//part b

//part c

else if (part == 'c')

{

cout << "Enter the maximum range under which you want to get twin primes " << endl;

long long range;

cin >> range;

twinprime(range);

}

//partc

}

//problem 1

//Problem 2

else if (problem == 2)

{

cout << "Enter a to check first part of the question otherwise enter b to check second part" << endl;

char choice;

cin >> choice;

//part a

system("cls");

if (choice == 'a'){

system("cls");

cout << "Enter a b c and d" << endl;

long long a, b, c, d;

cout << "Enter a:"; cin >> a;

cout << "Enter b:"; cin >> b;

cout << "Enter c:"; cin >> c;

cout << "Enter d:"; cin >> d;

system("cls");

if (EularConjucture(a, b, c, d) == true)

{

cout << "Conjucture satisfied" << endl;

cout << "a=" << a << endl;

cout << "b=" << b << endl;

cout << "c=" << c << endl;

cout << "d=" << d << endl;

}

else if (EularConjucture(a, b, c, d) == false)

{

cout << "Conjucture Failed" << endl;

cout << "a=" << a << endl;

cout << "b=" << b << endl;

cout << "c=" << c << endl;

cout << "d=" << d << endl;

}

}

//part a

//part b

else if (choice == 'b')

{

long long range = 2147383647;

for (int a = range; a>0; a--)

{

for (int b = range; b > 0; b--)

{

for (int c = range; c > 0; c--)

{

for (int d = range; d > 0; d--)

{

if (EularConjucture(a, b, c, d) == true)

{

cout << "Finally" << endl;

cout << "Conjucture satisfied" << endl;

cout << "a=" << a << endl;

cout << "b=" << b << endl;

cout << "c=" << c << endl;

cout << "d=" << d << endl;

}

}

}

}

}

// 2682440, 15365639, 18796760, 20615673. Solution after long term testing.

}

//part b

}

//problem2

//problem 3

else if (problem == 3)

{

cout << "Enter The part of the question you want to check (a,b,c)" << endl;

char part;

cin >> part;

system("cls");

//part a

if (part == 'a')

{

cout << "Press e to enter triptlet to check if it satisfies 313(a^3+b^3)=c^3.\n\nElse press t to find 3 tuple which satisfies given proposition:" << endl;

char option;

cin >> option;

system("cls");

if (option == 'e')

{

cout << "Enter a,b and c" << endl;

long long a, b, c;

cout << "Enter a:"; cin >> a;

cout << "Enter b:"; cin >> b;

cout << "Enter c:"; cin >> c;

system("cls");

if (ellipticcurve(a, b, c) == true)

{

cout << "Conjucture satisfied" << endl;

cout << "a=" << a << endl;

cout << "b=" << b << endl;

cout << "c=" << c << endl;

}

else if (ellipticcurve(a, b, c) == false)

{

cout << "Conjucture Failed" << endl;

cout << "a=" << a << endl;

cout << "b=" << b << endl;

cout << "c=" << c << endl;

}

}

else if (option == 't')

{

long long range = 2147383647;

for (int a = range; a > 0; a--)

{

for (int b = range; b > 0; b--)

{

for (int c = range; c > 0; c--)

{

if (ellipticcurve(a, b, c) == true)

{

cout << "Finally!!" << endl;

cout << "Conjucture satisfied" << endl;

cout << "a=" << a << endl;

cout << "b=" << b << endl;

cout << "c=" << c << endl;

}

}

}

}

}

}

//part b

else if (part == 'b')

{

cout << "How many numbers you want to test :" << endl;

int limit;

cin >> limit;

system("cls");

for (int start = 0; start < limit; start++)

{

cout << "Enter only the even number you want to test: " << endl;

long long number;

cin >> number;

goldbach(number);

cout << endl;

}

}

//part b

else if (part == 'c')

{

cout << "How many numbers you want to test :" << endl;

int limit;

cin >> limit;

system("cls");

for (int start = 0; start < limit; start++)

{

cout << "Enter number you want to test: " << endl;

long long number;

cin >> number;

Collatz(number);

cout << endl;

}

}

}

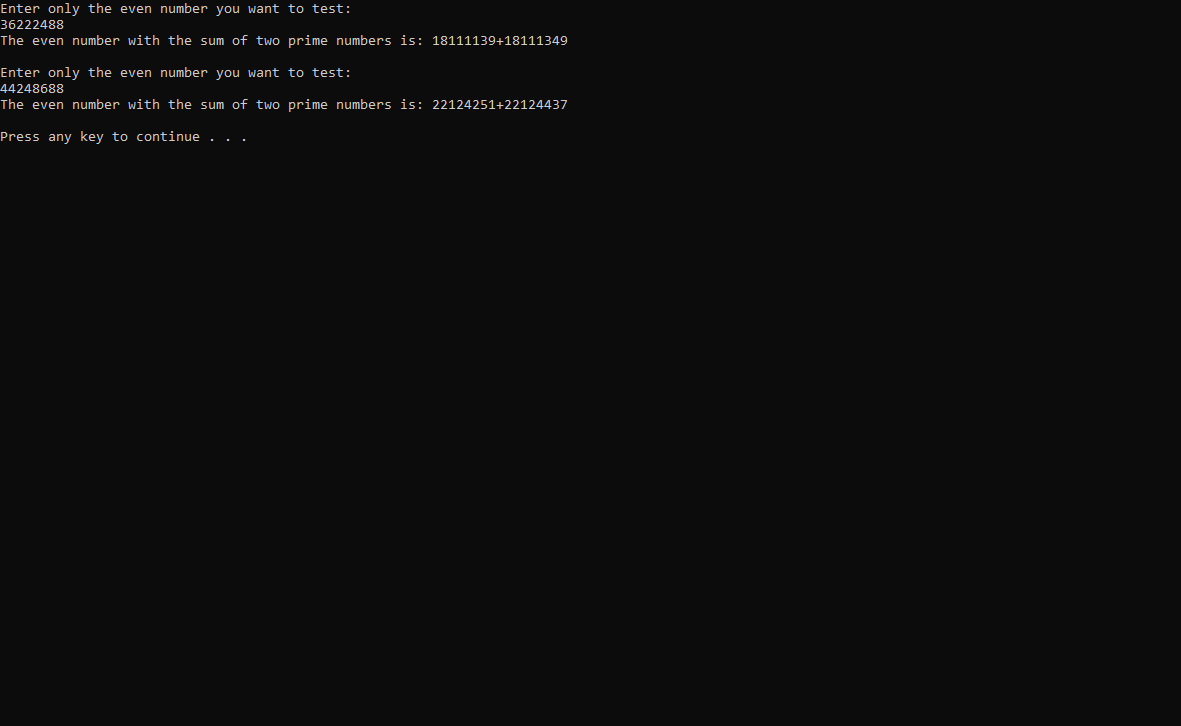
//problem 3

system("pause");

}

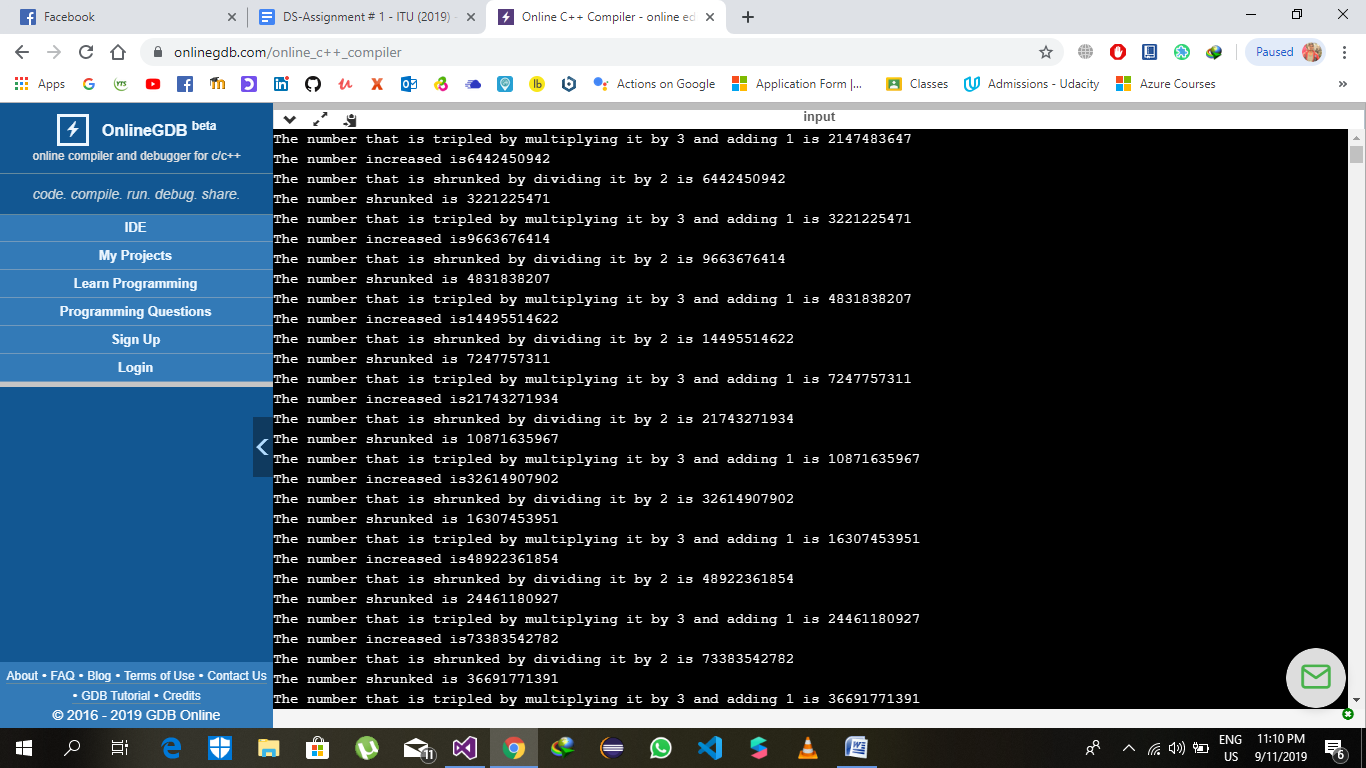
OUTPUTS

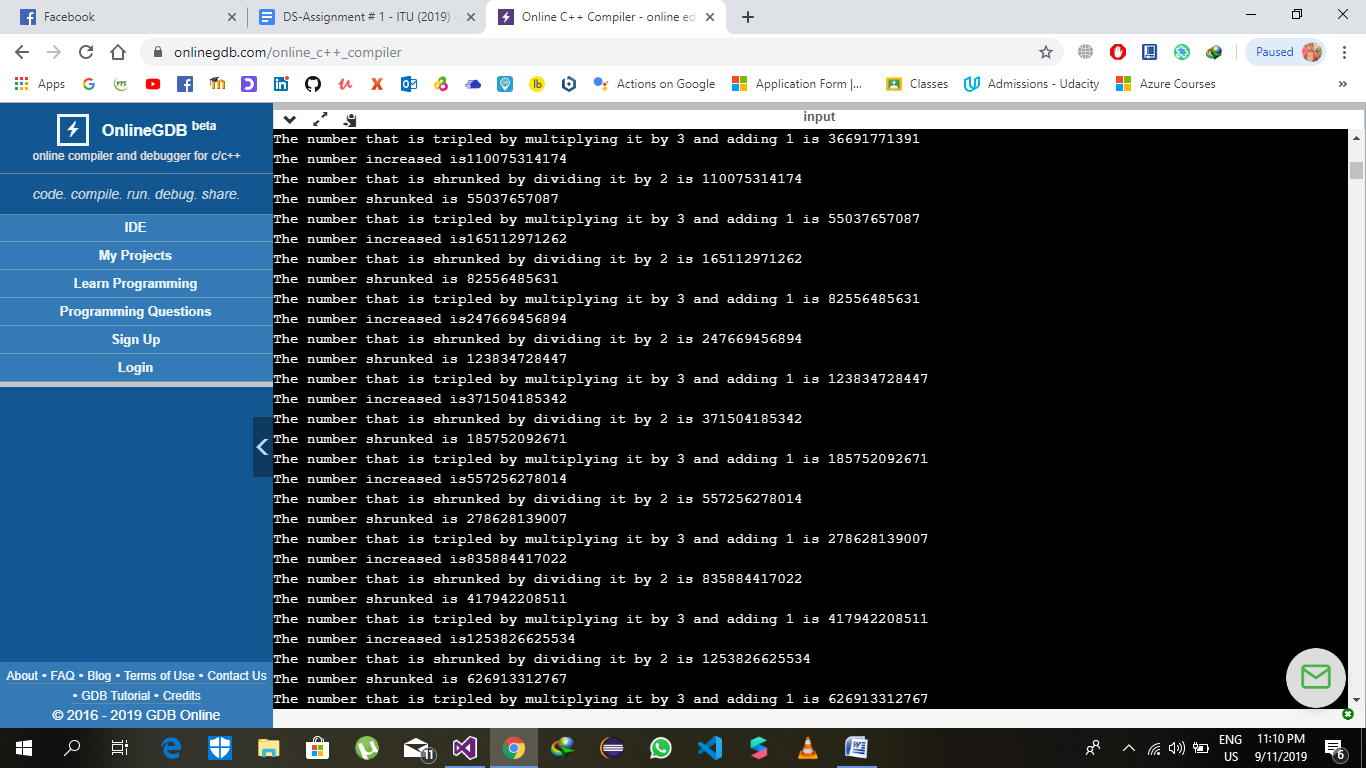
PROBLEM 3-b

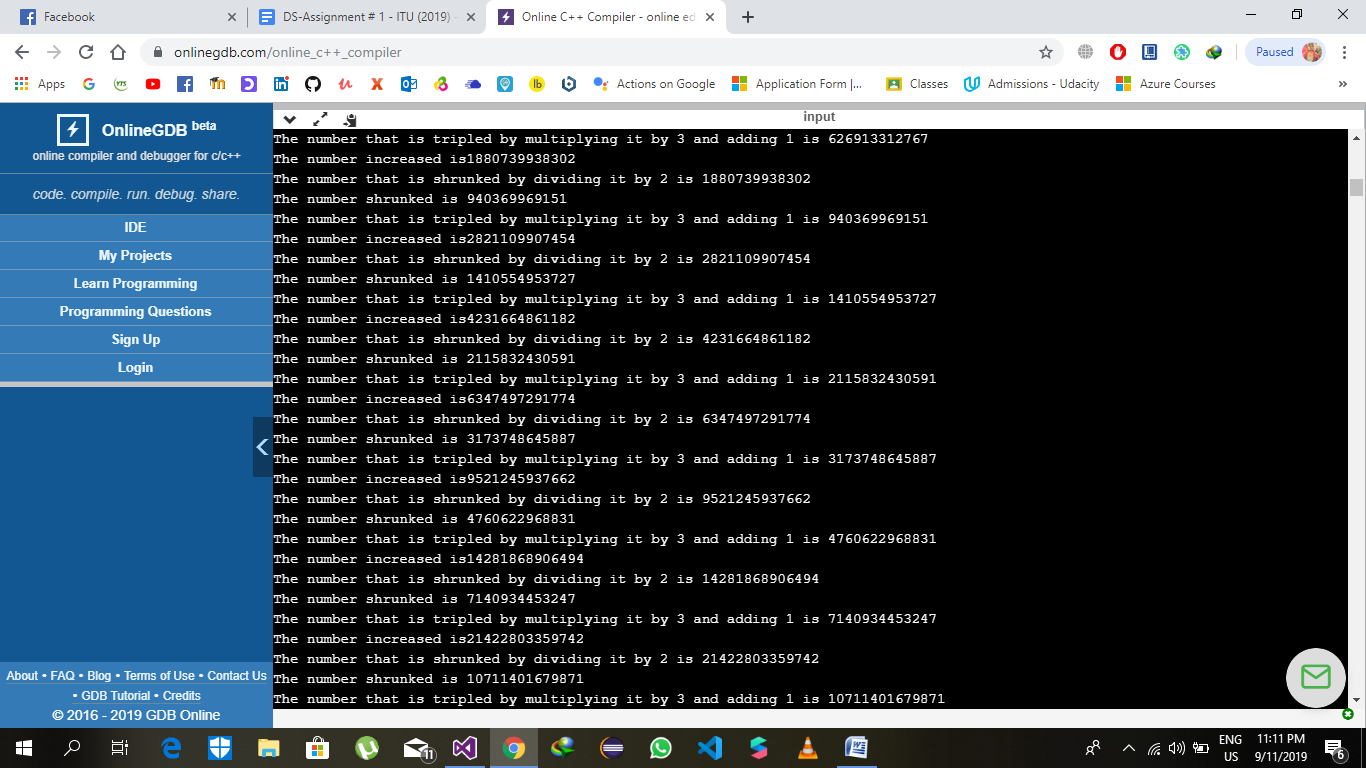


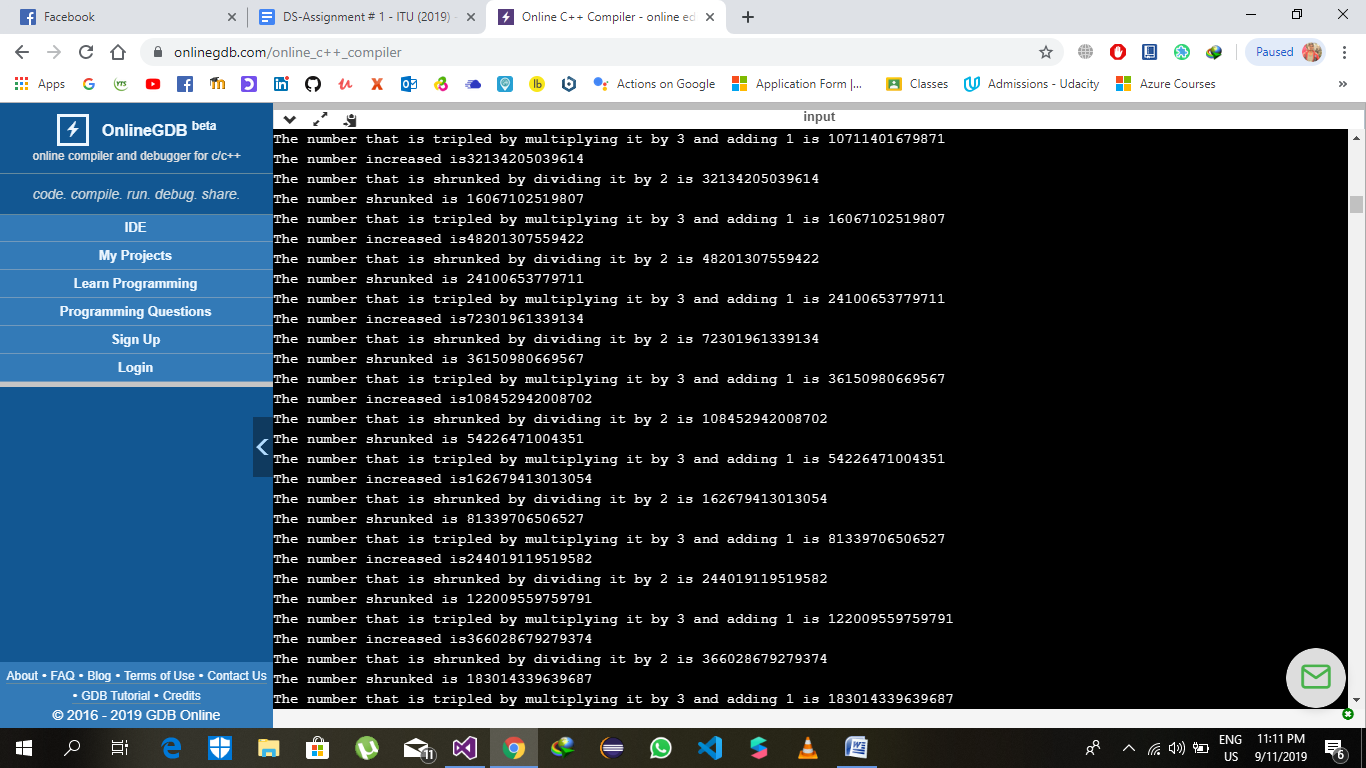
a)

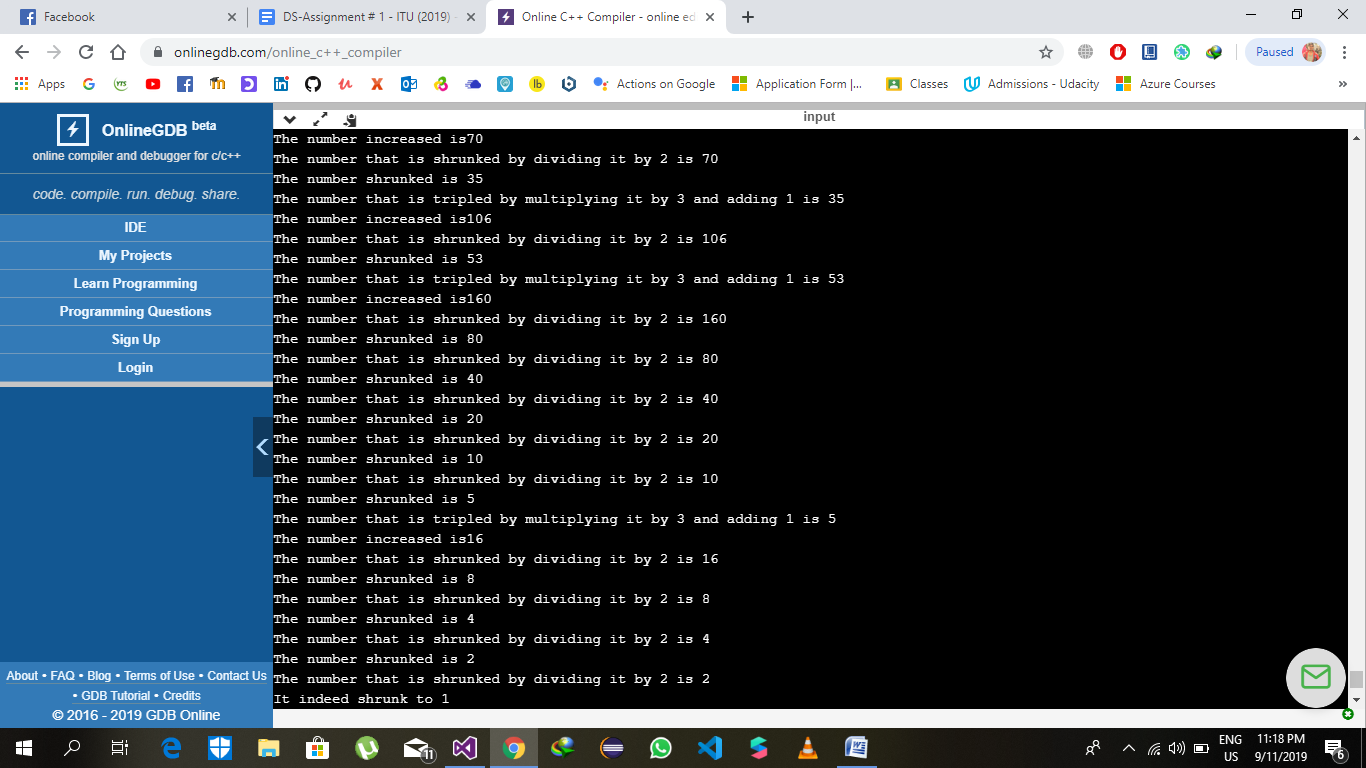
Number 1 = 2147483647

\

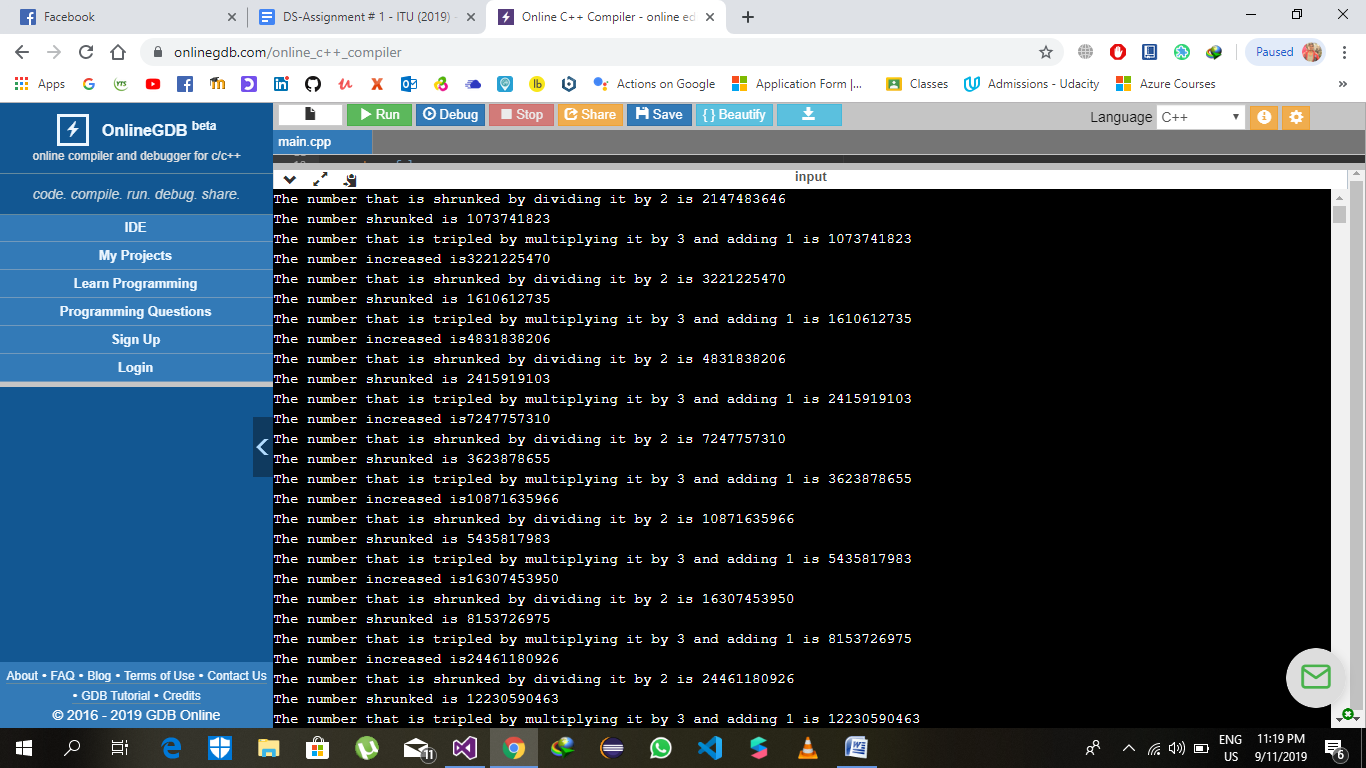


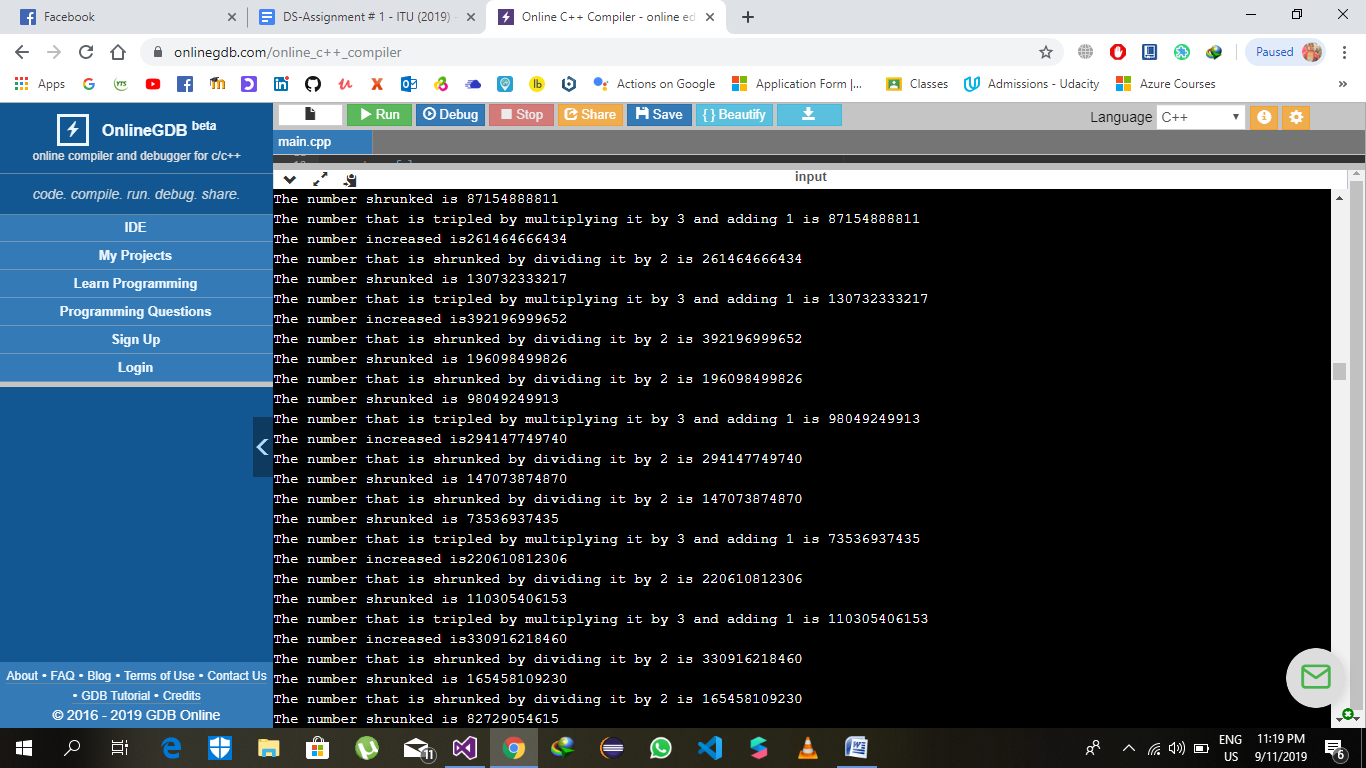


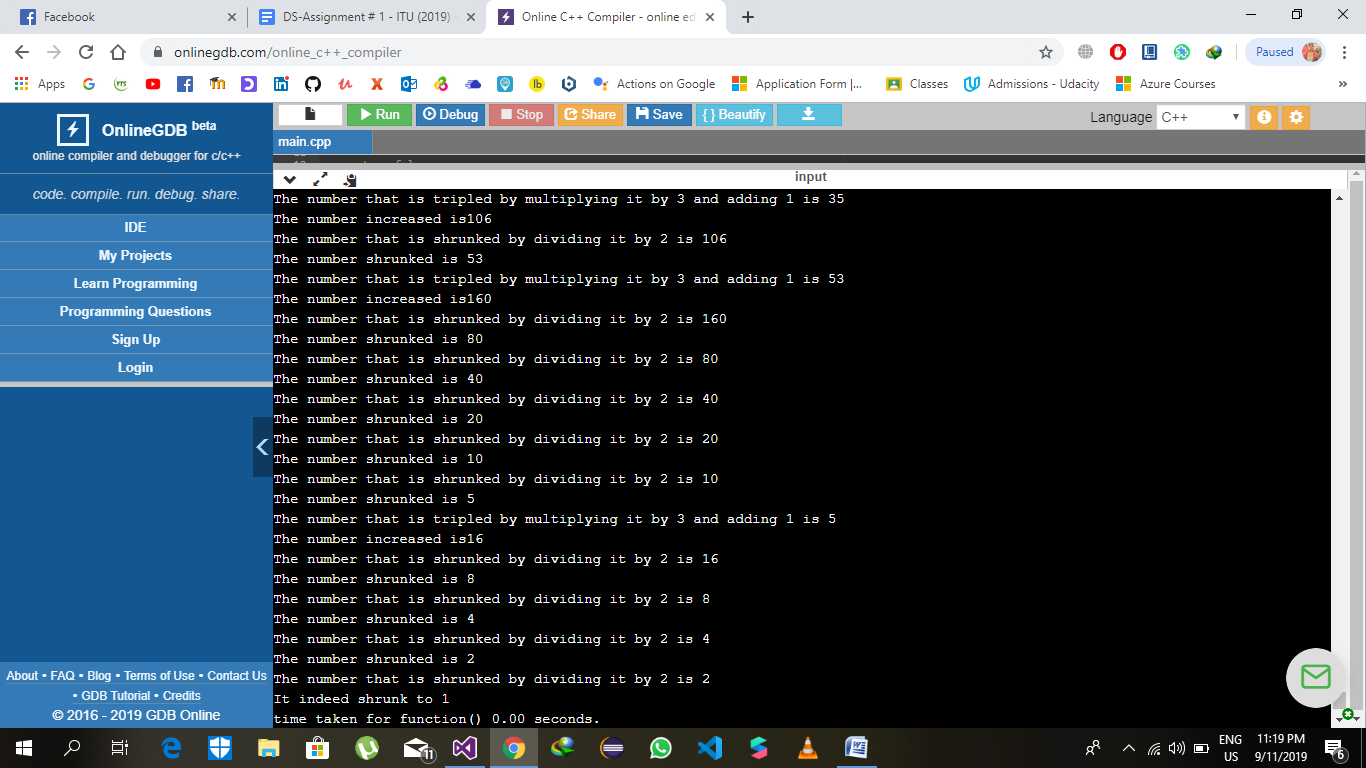




Number 2 = 2147483646

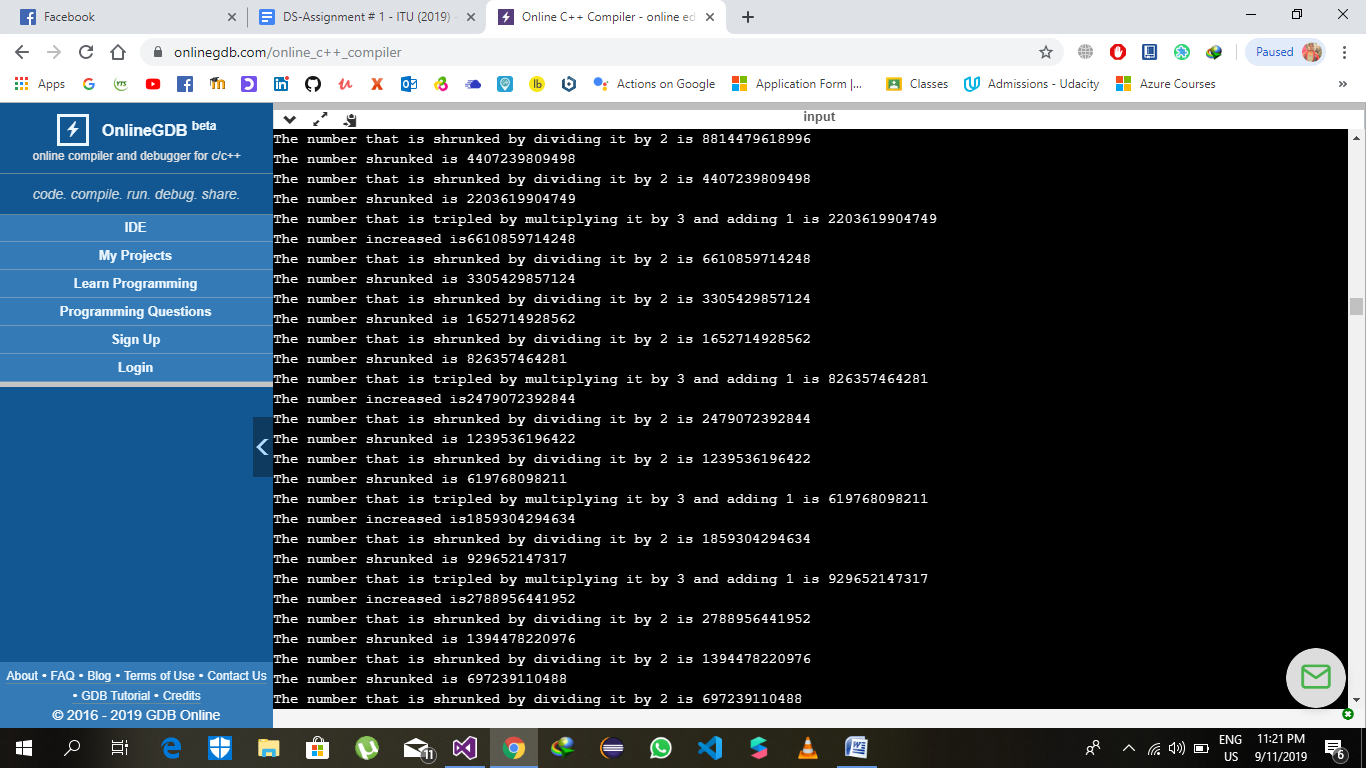


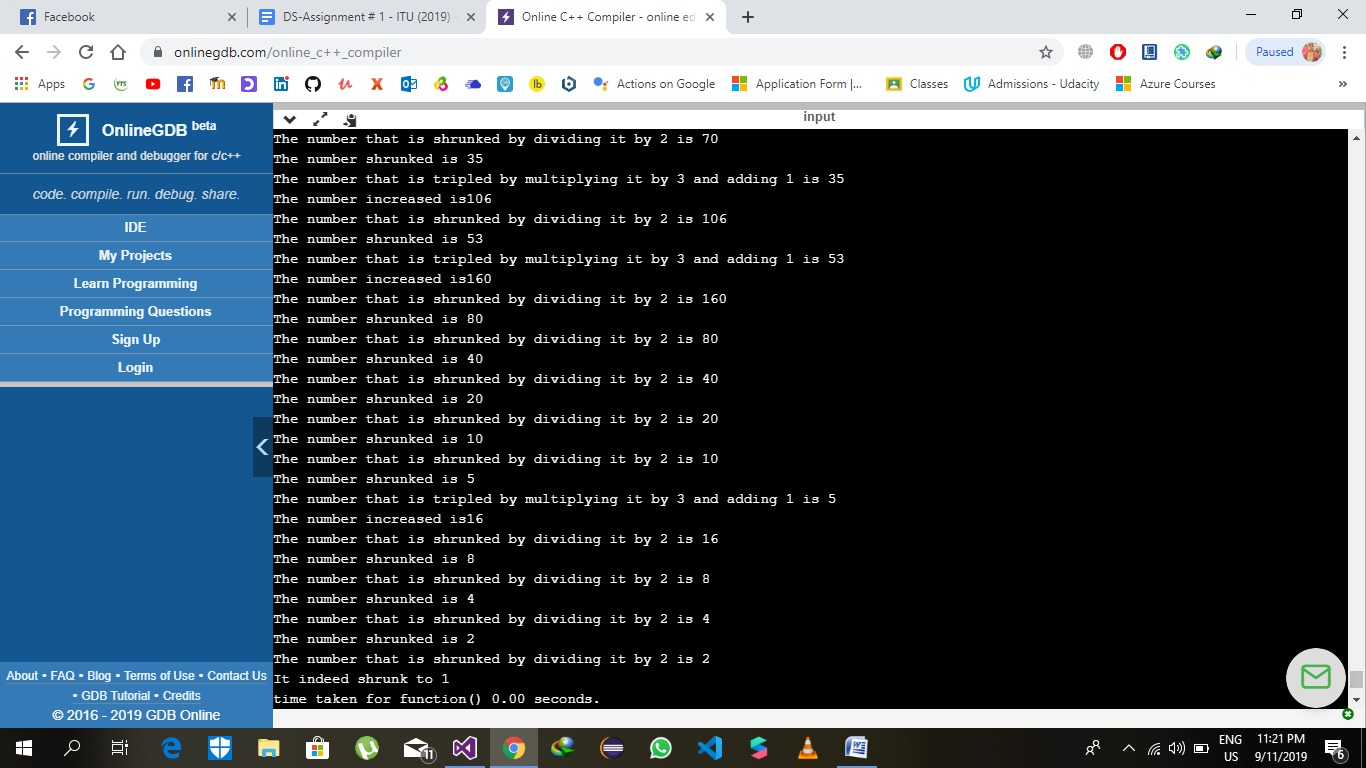




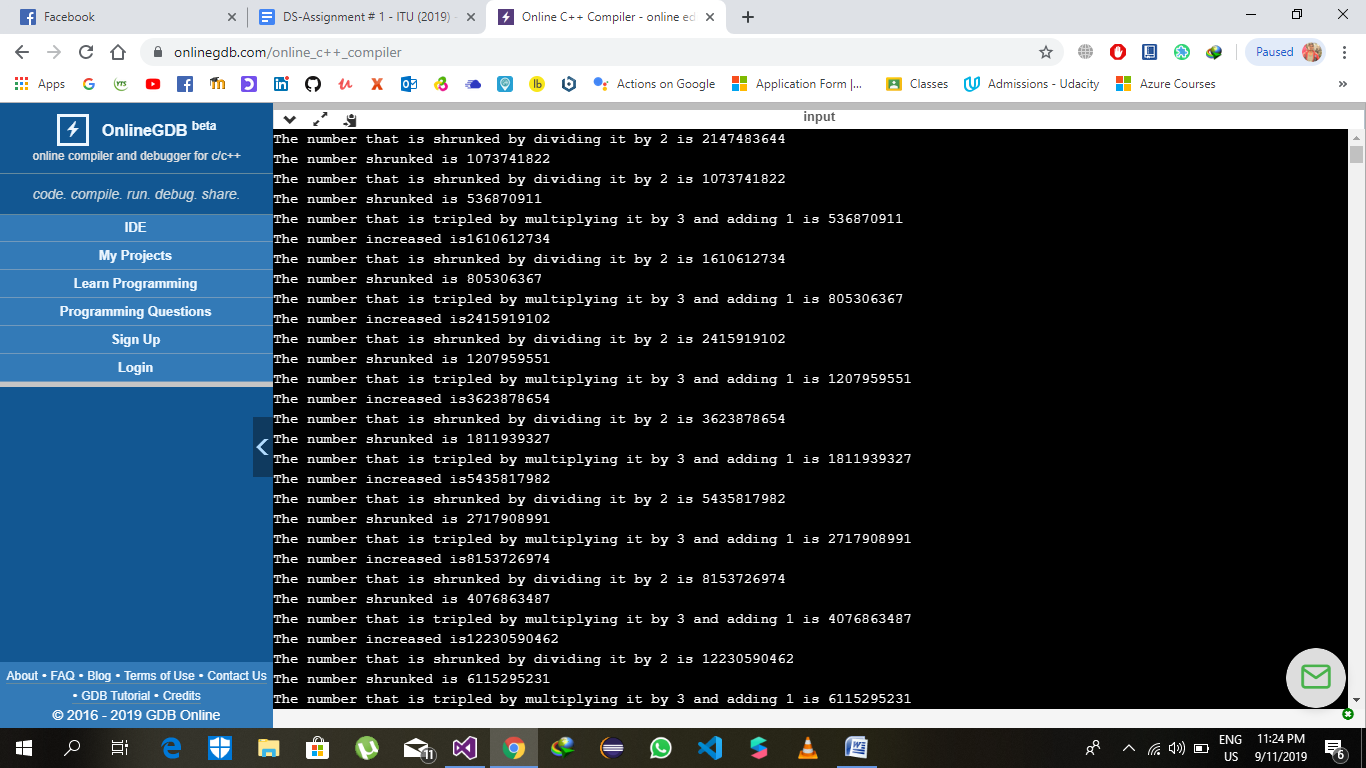
Number 3 = 2147483645

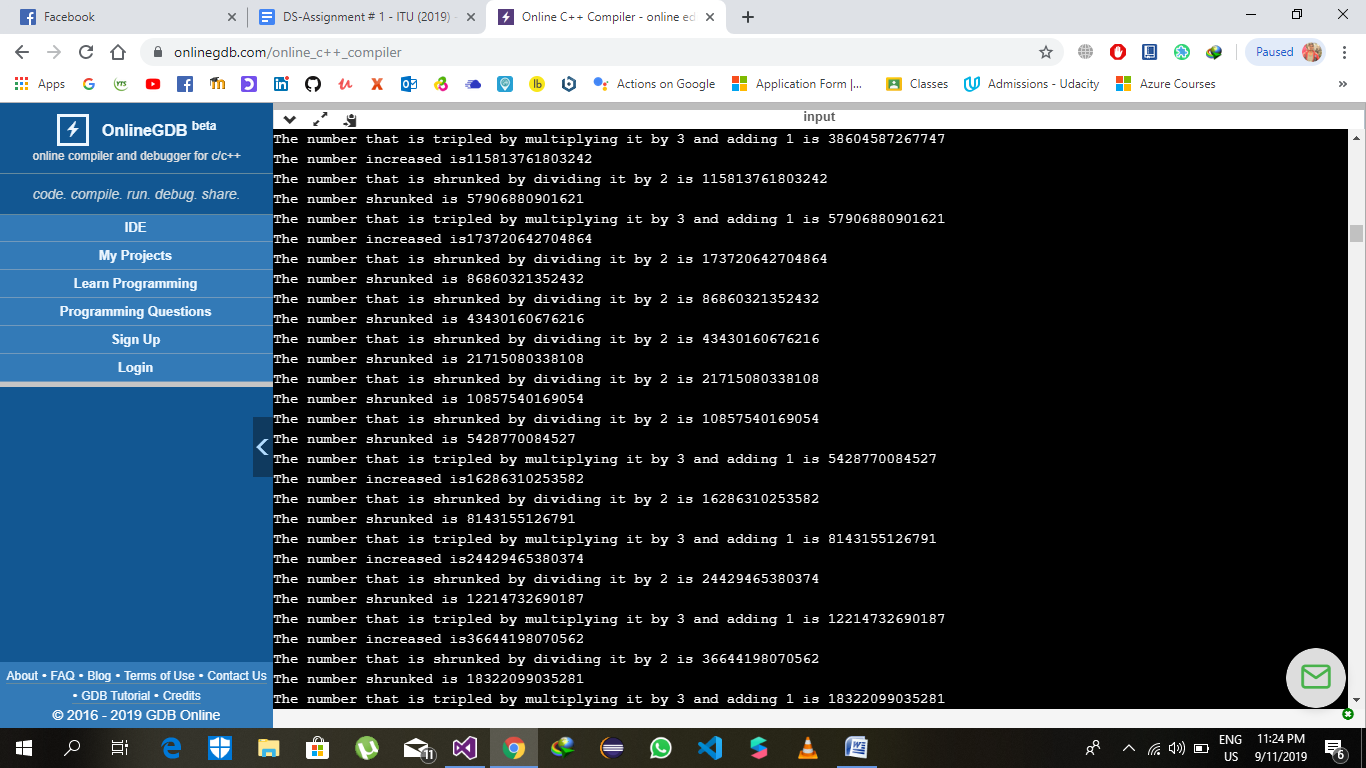


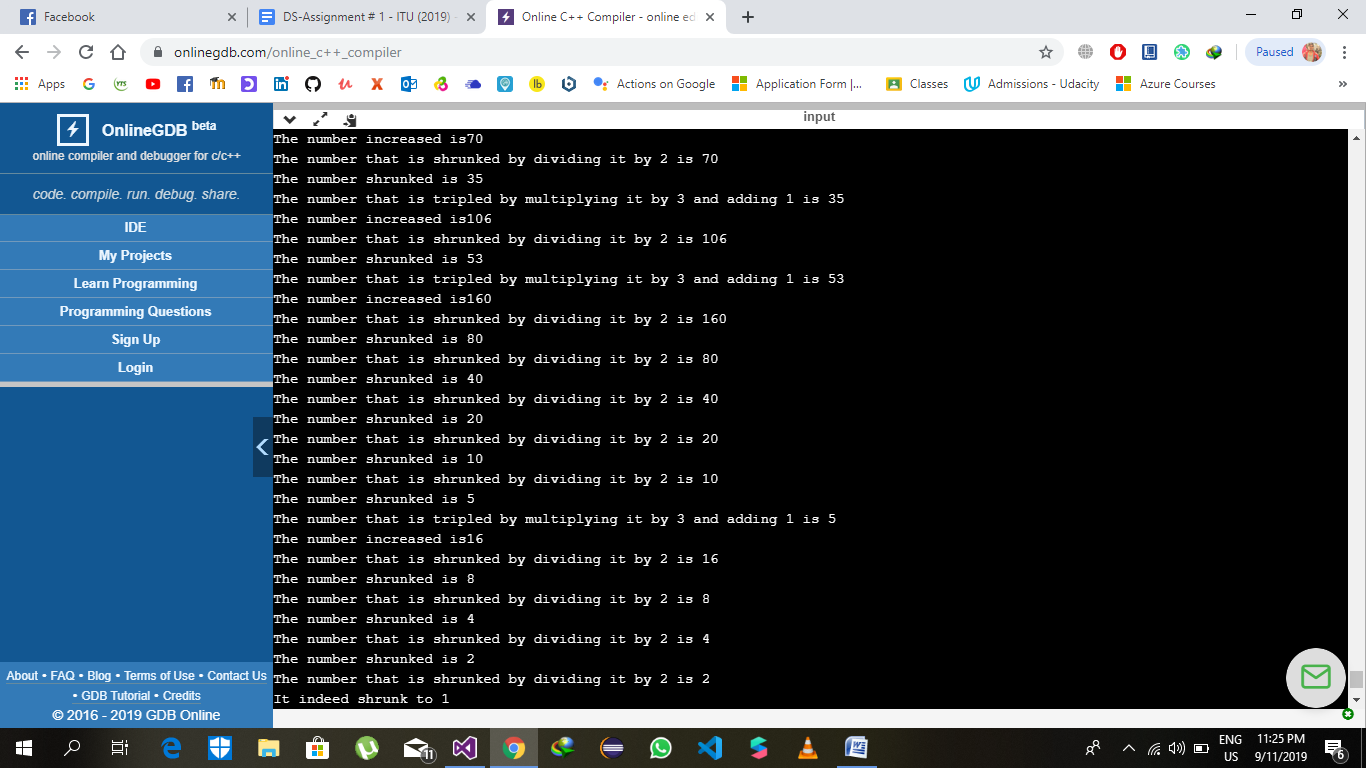




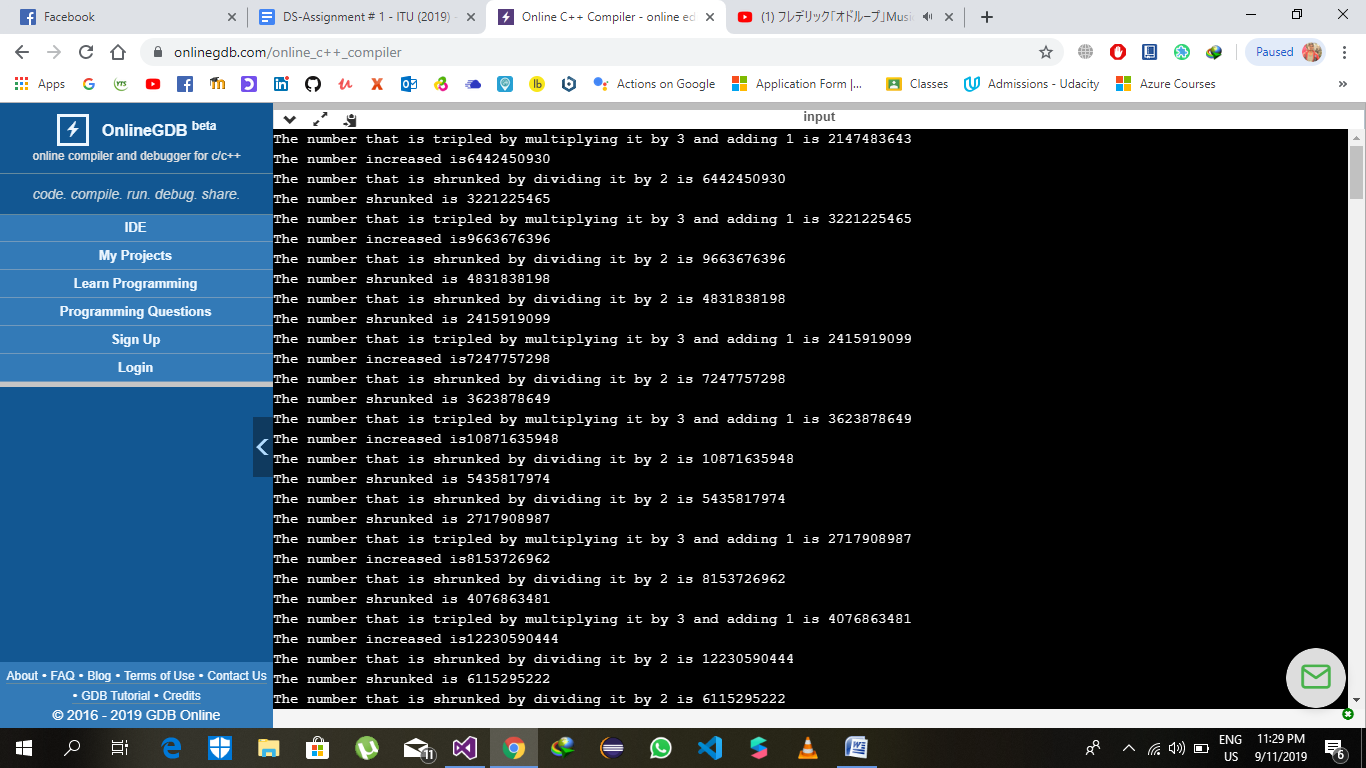
Number 4 = 2147483644



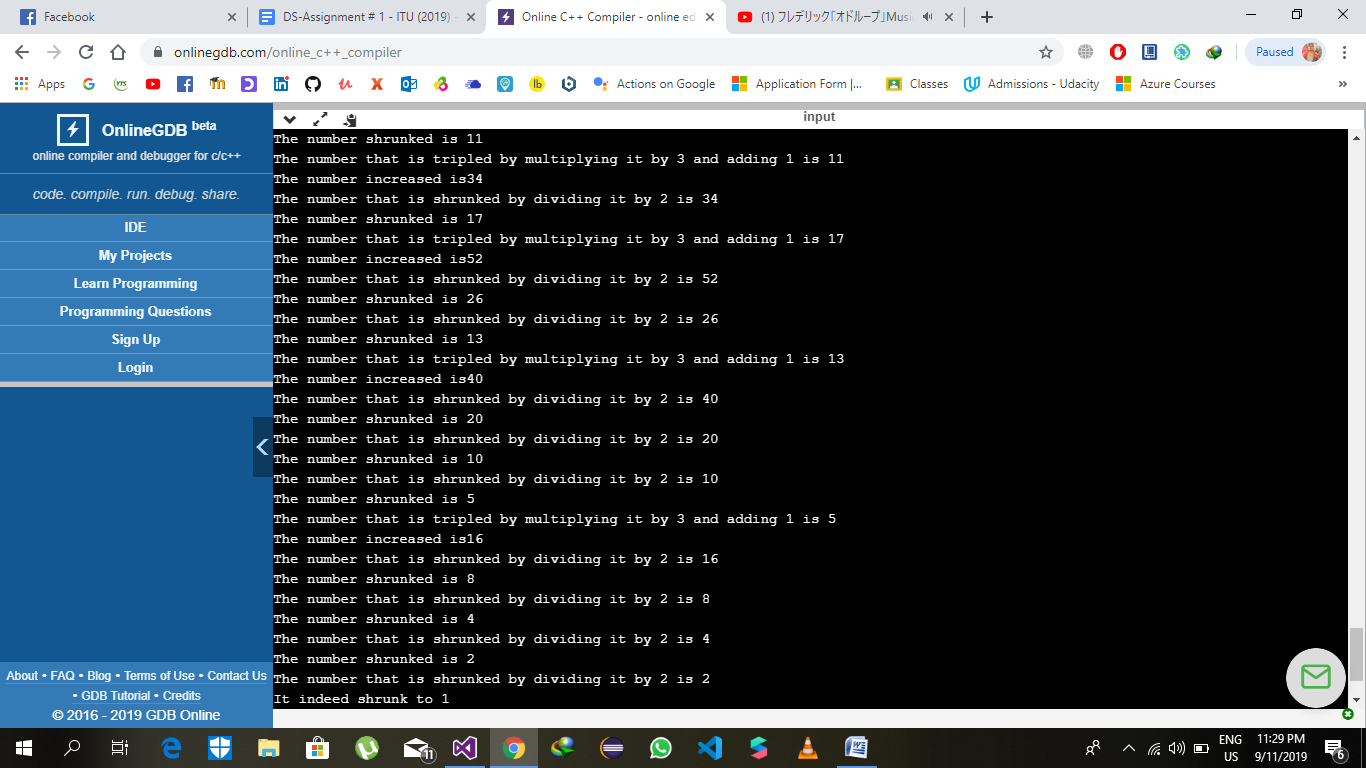




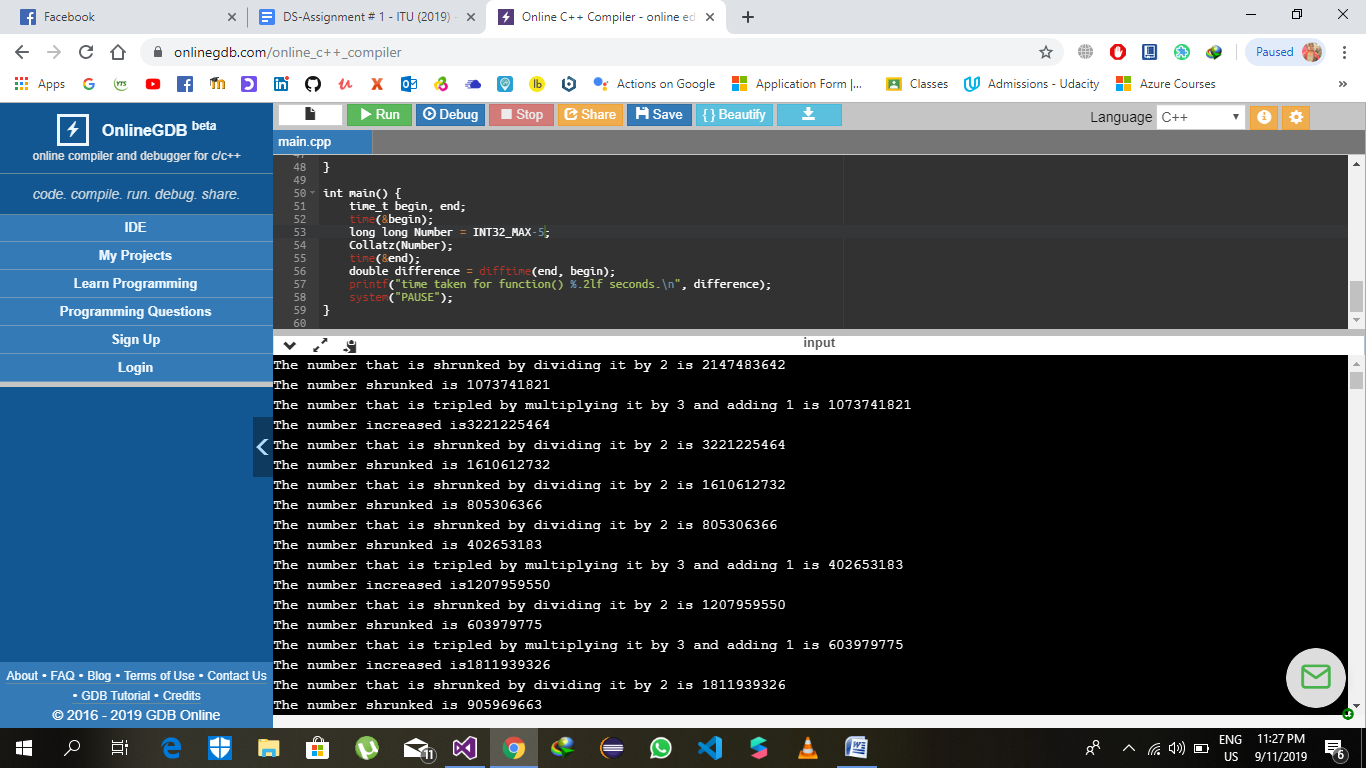
The number 5 = 2147483643

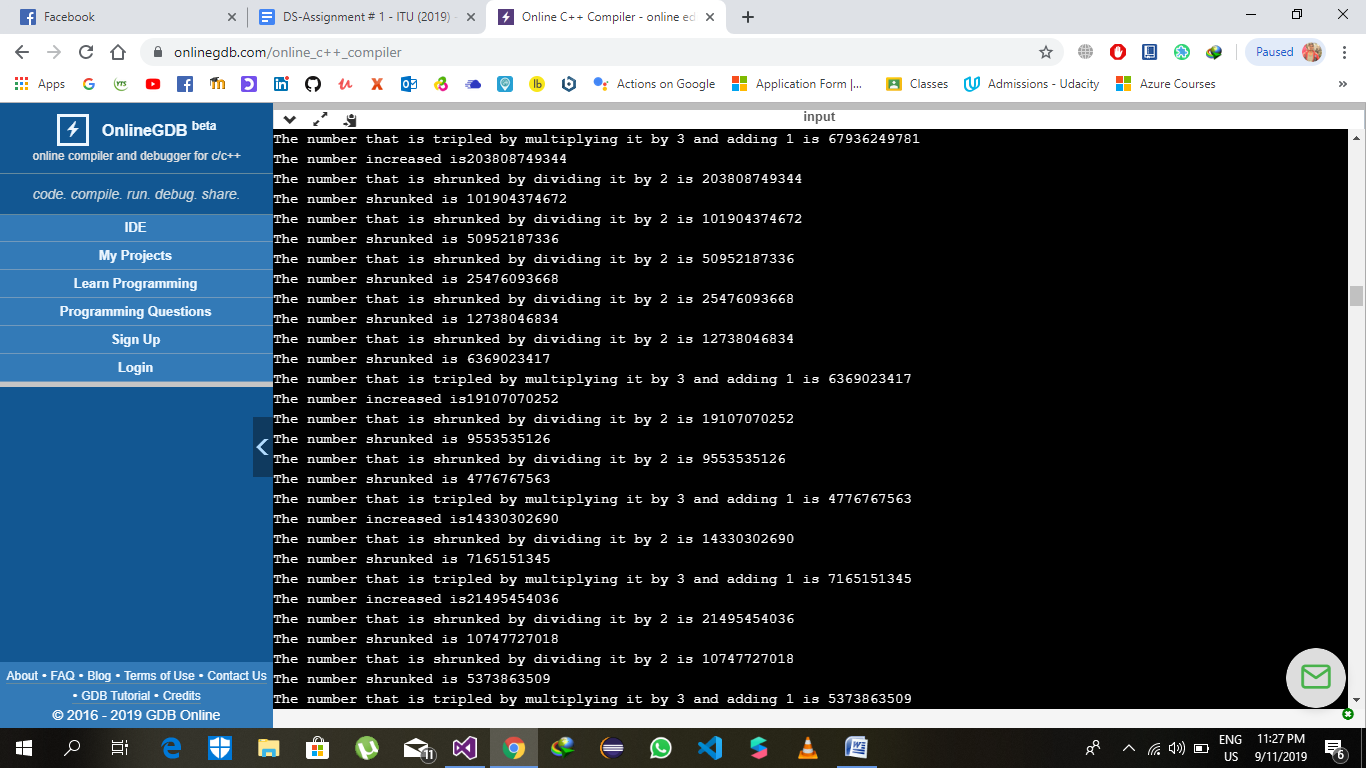


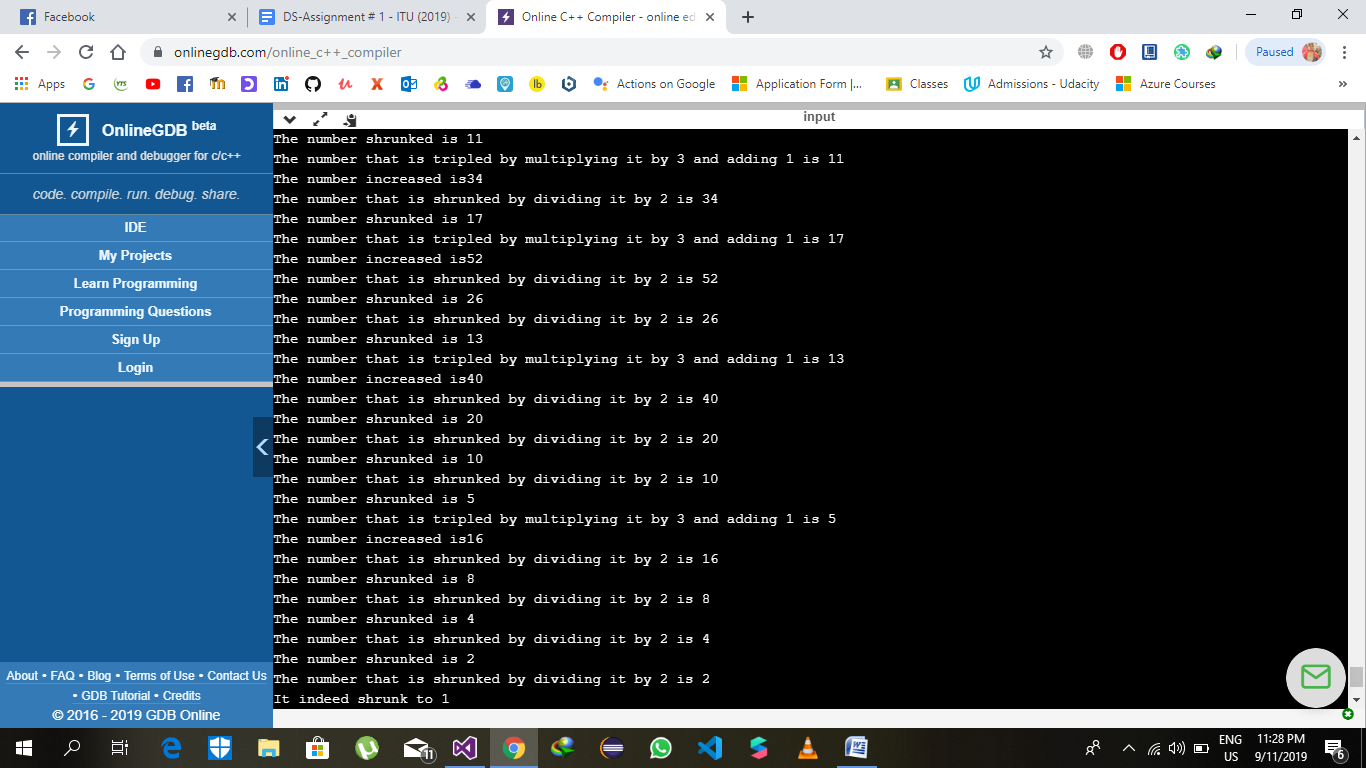




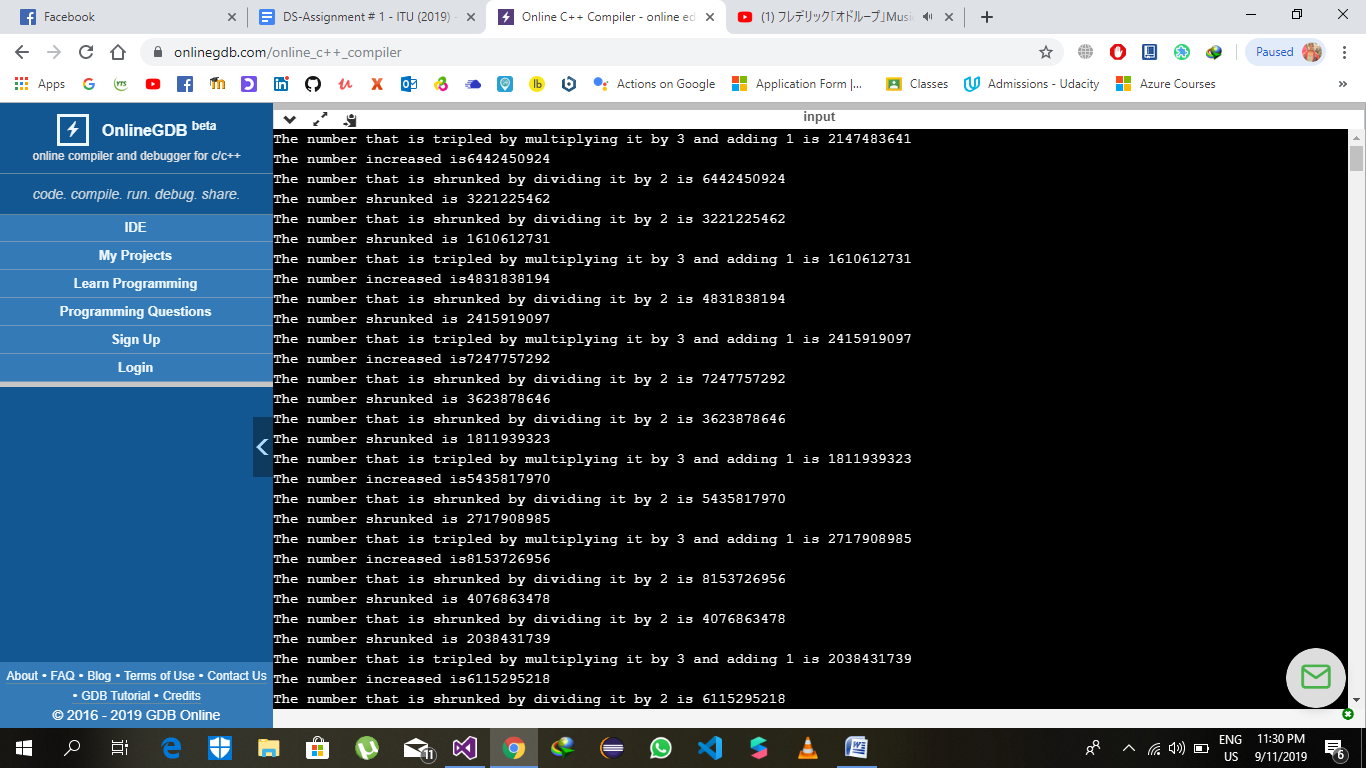
The number 6 = 2147483642

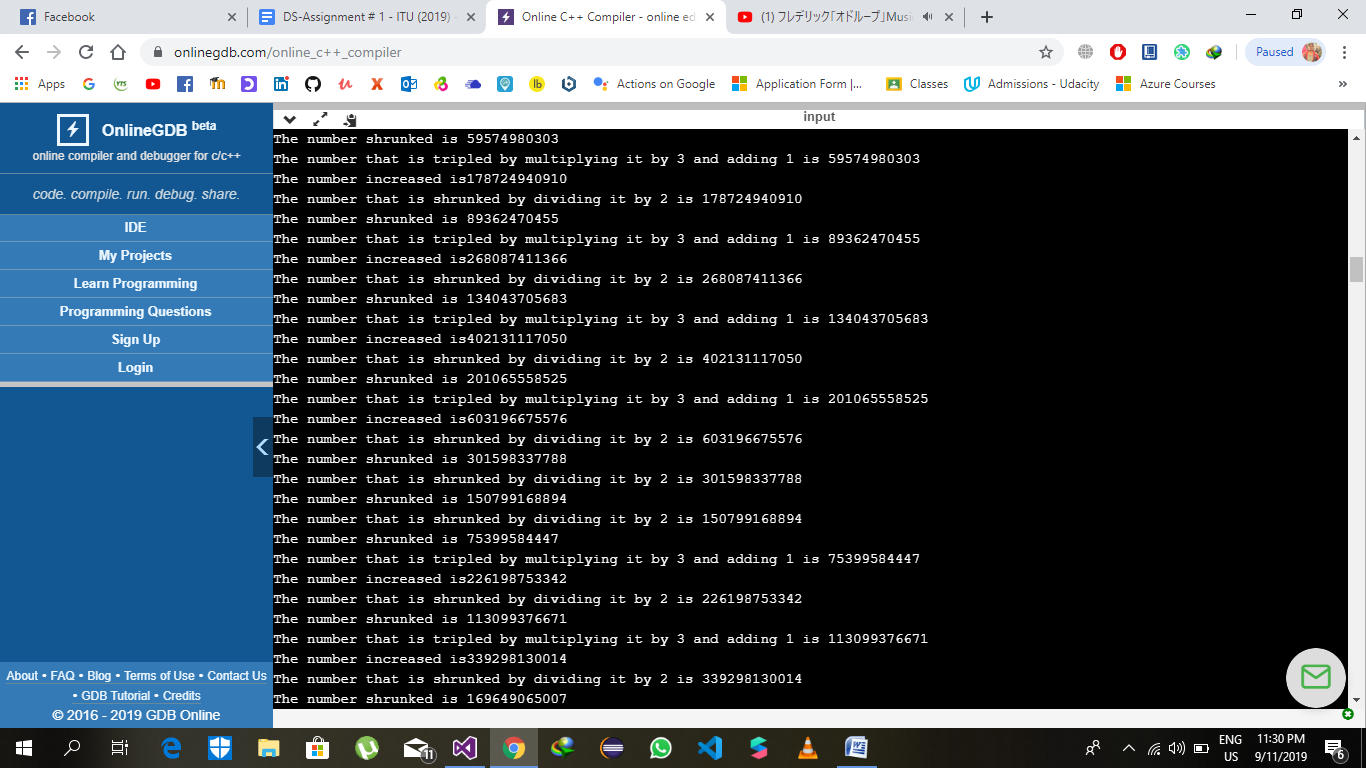


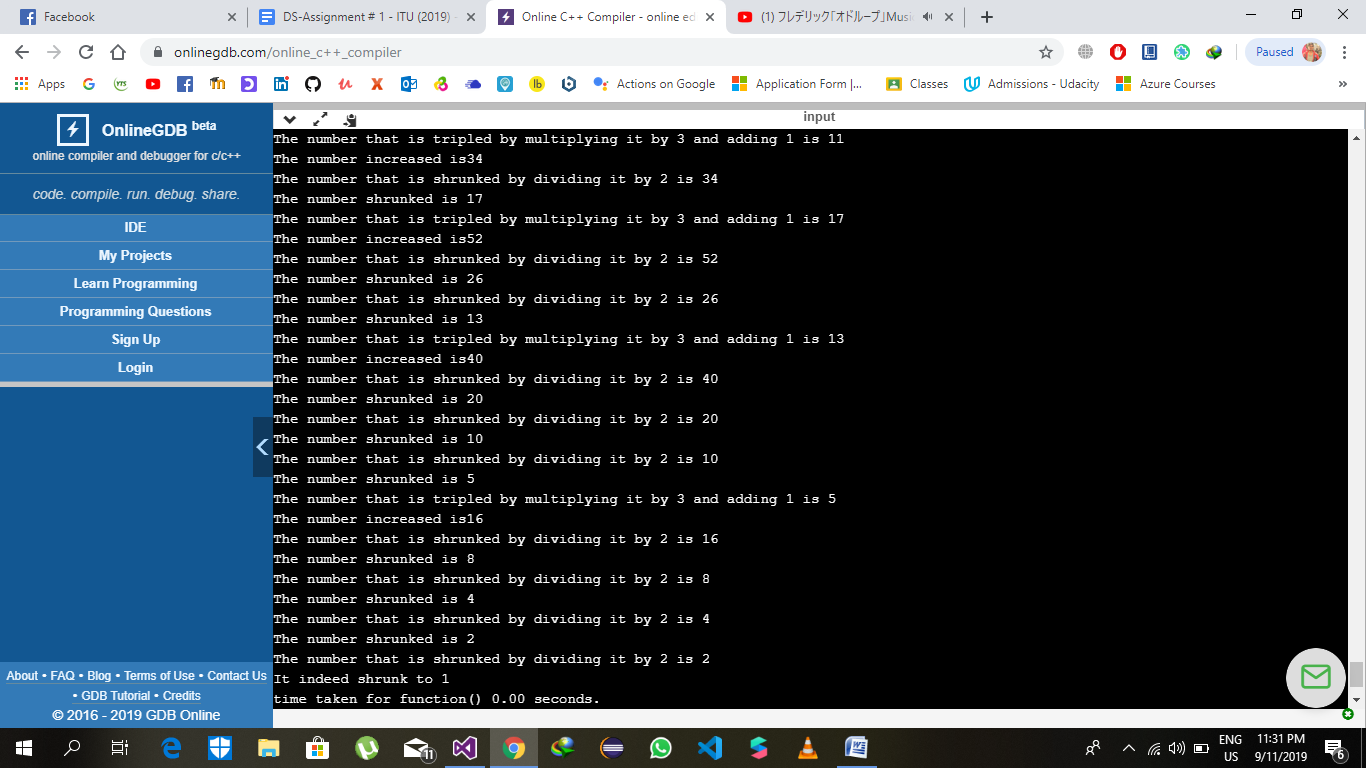




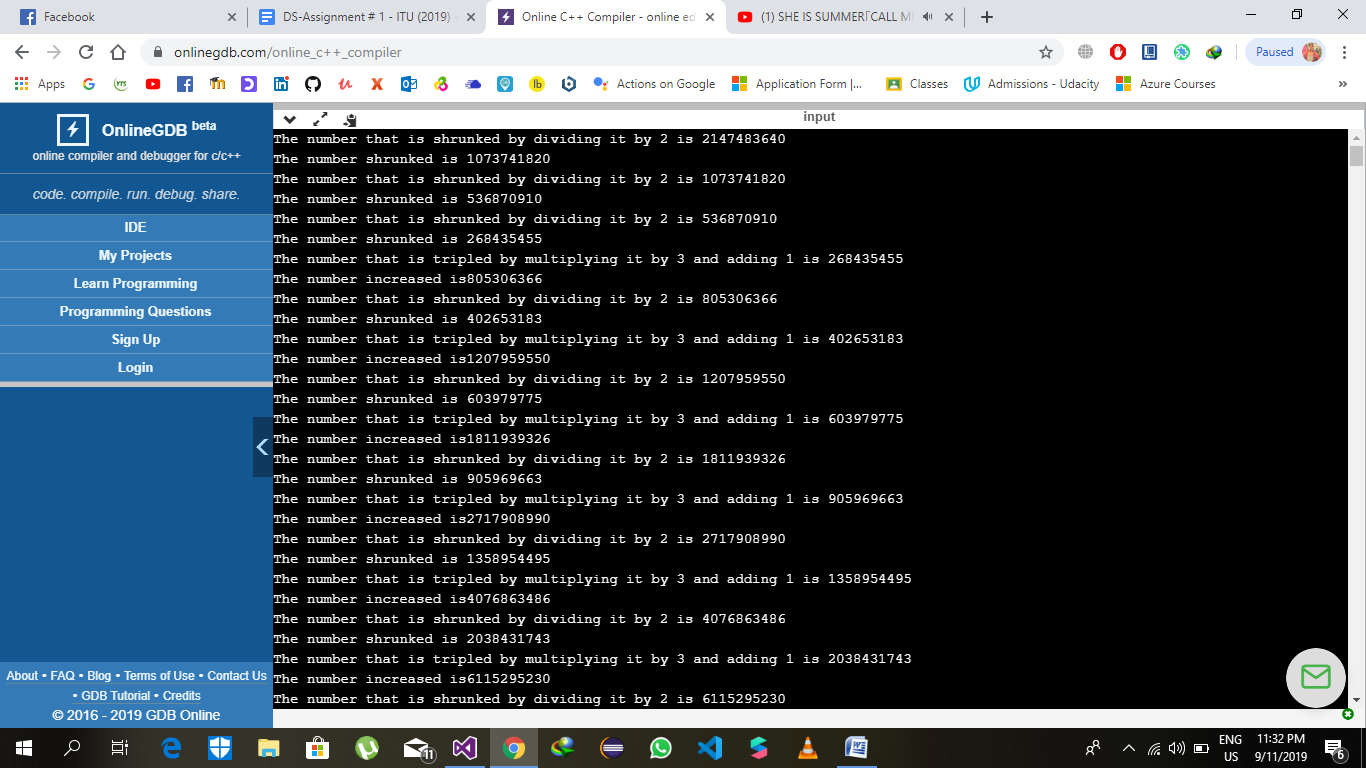
The number 7 = 2147483641

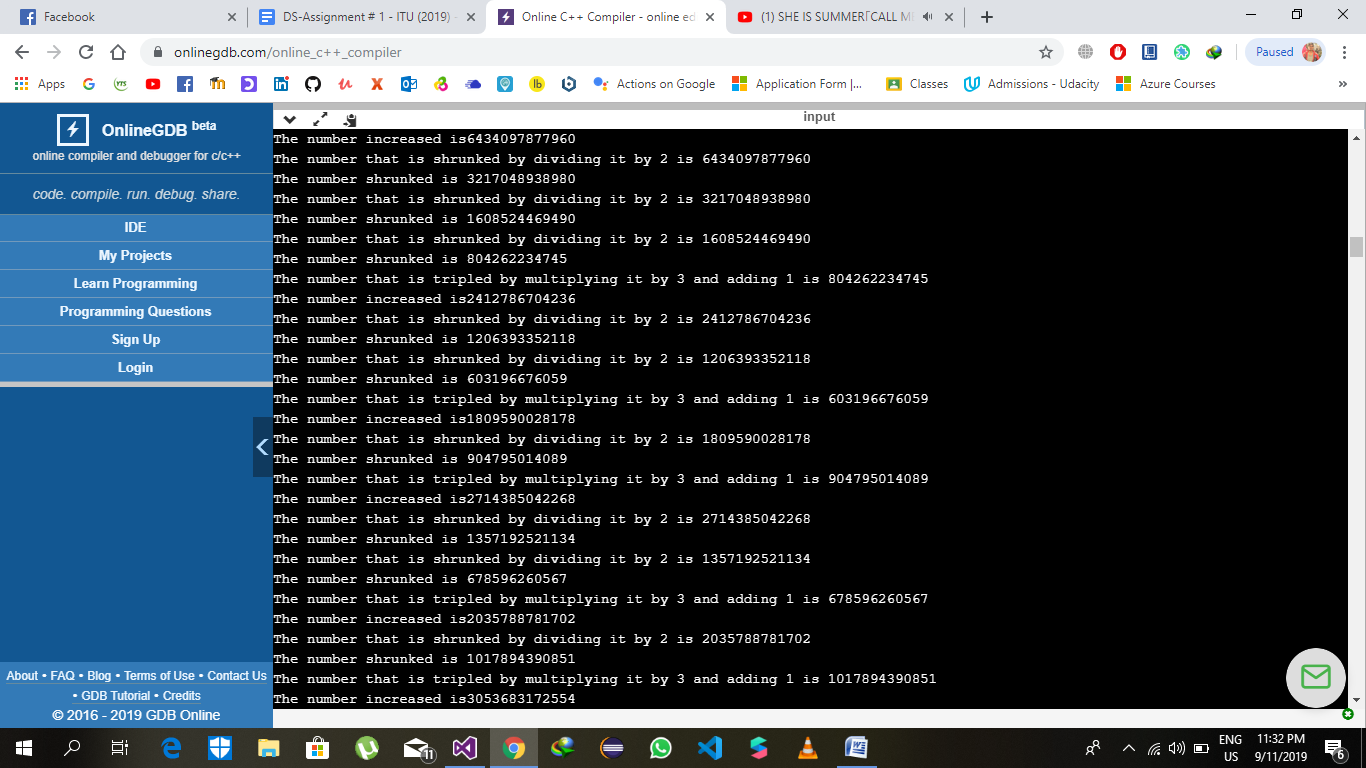


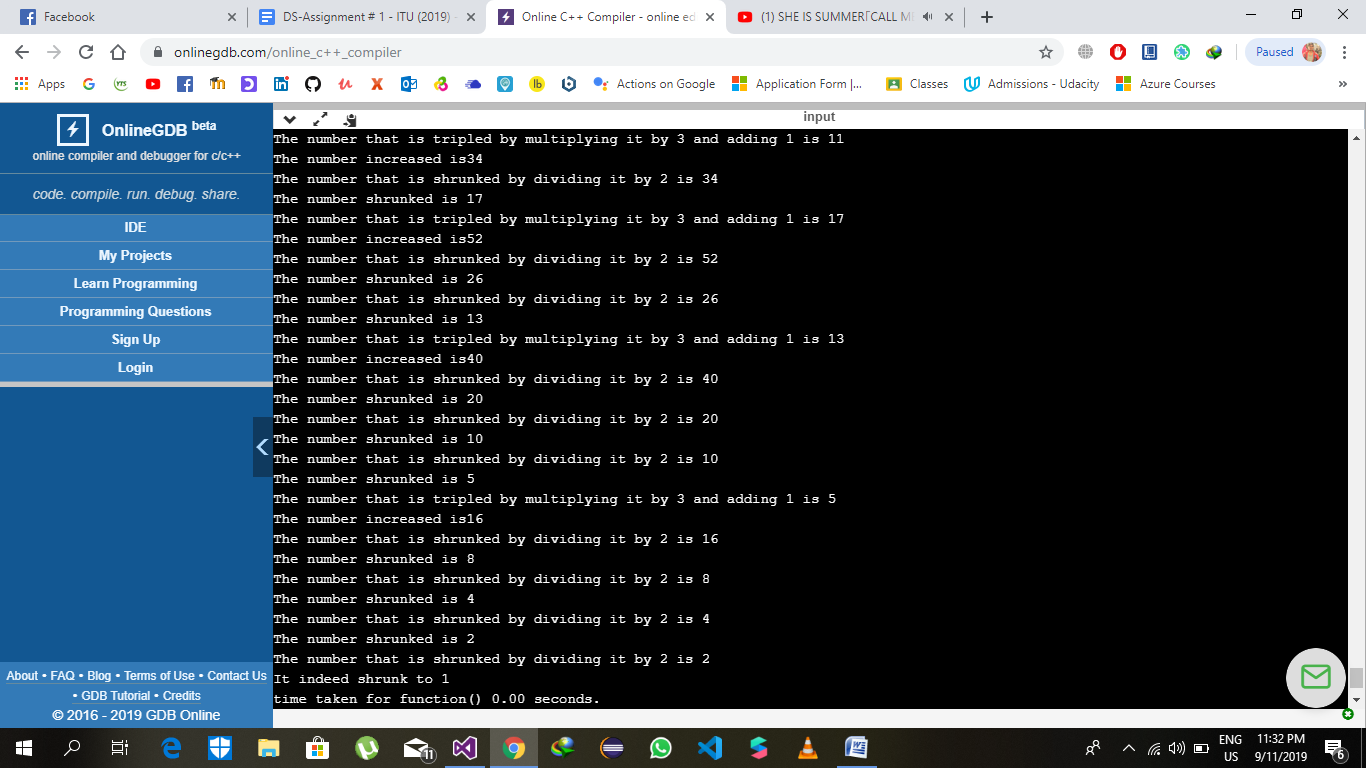




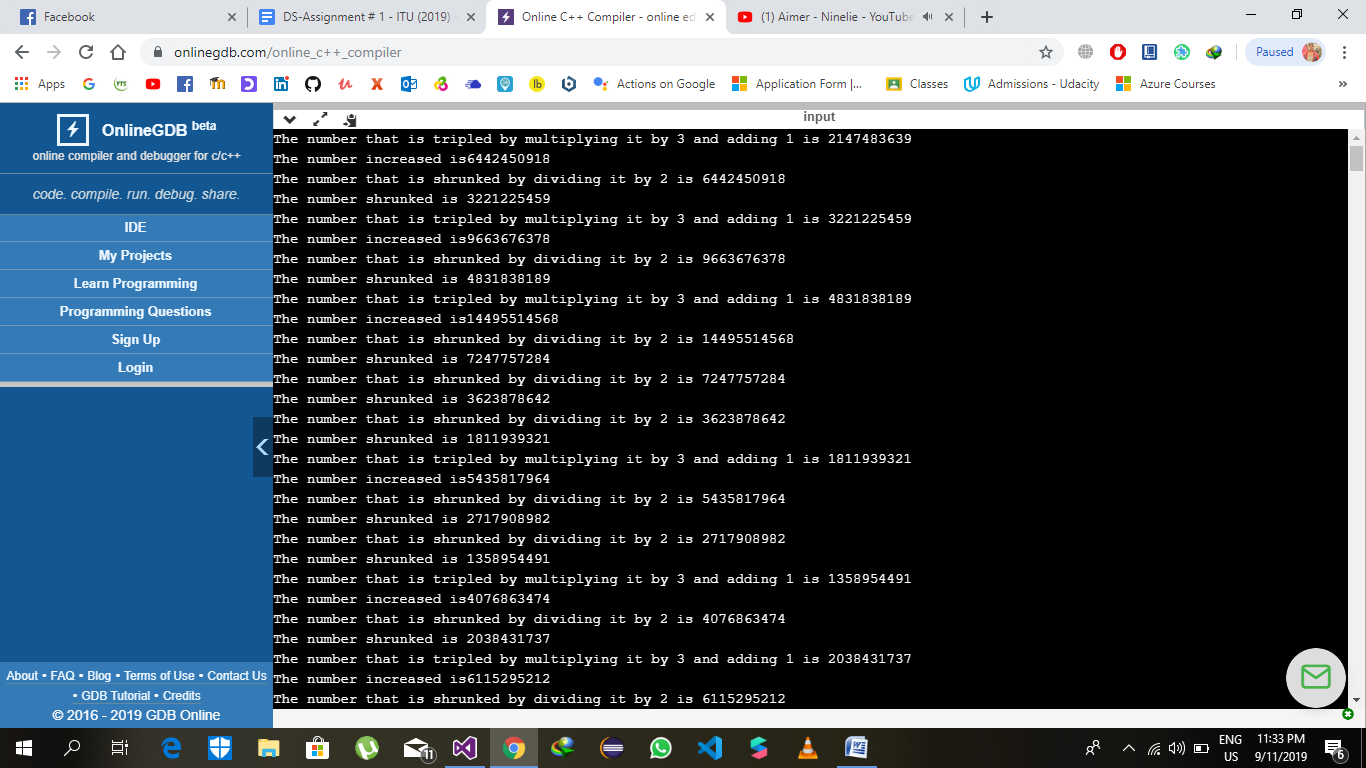
The number 8 = 2147483640

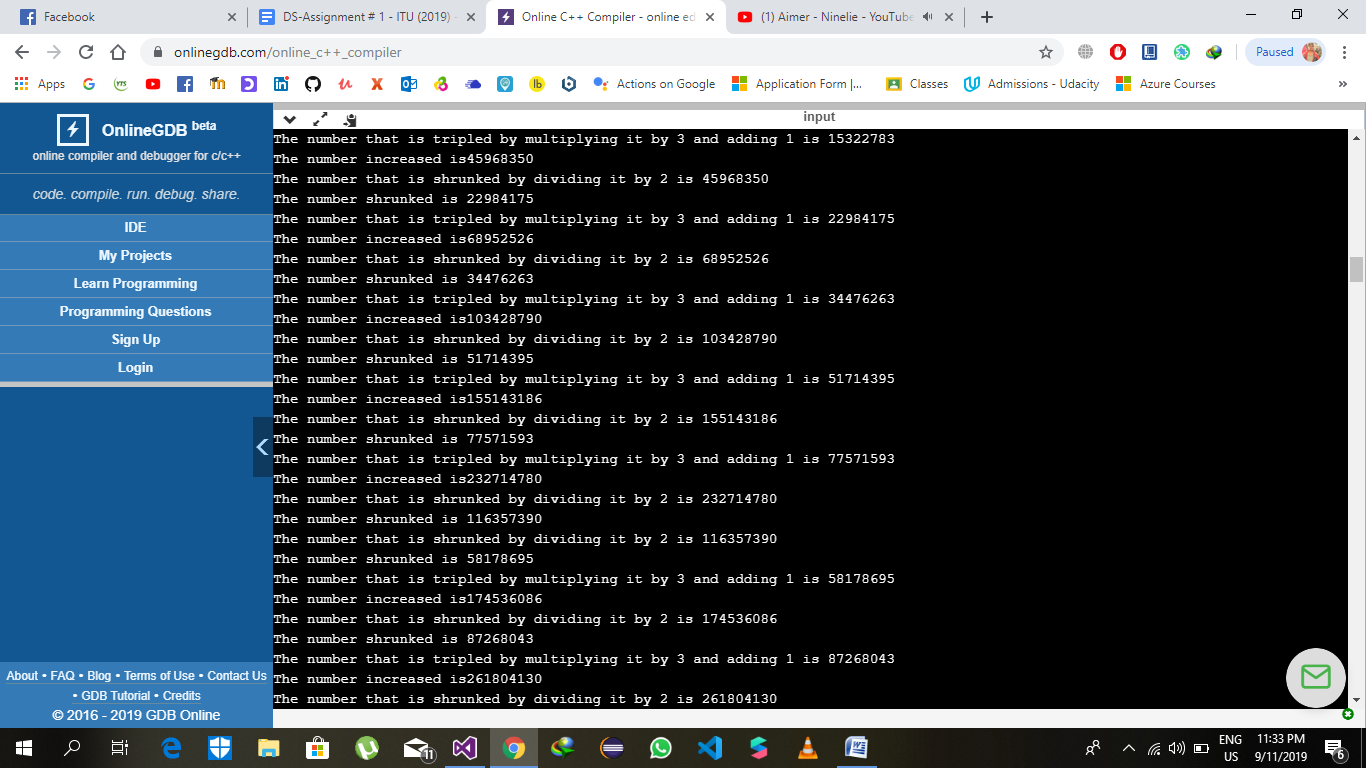


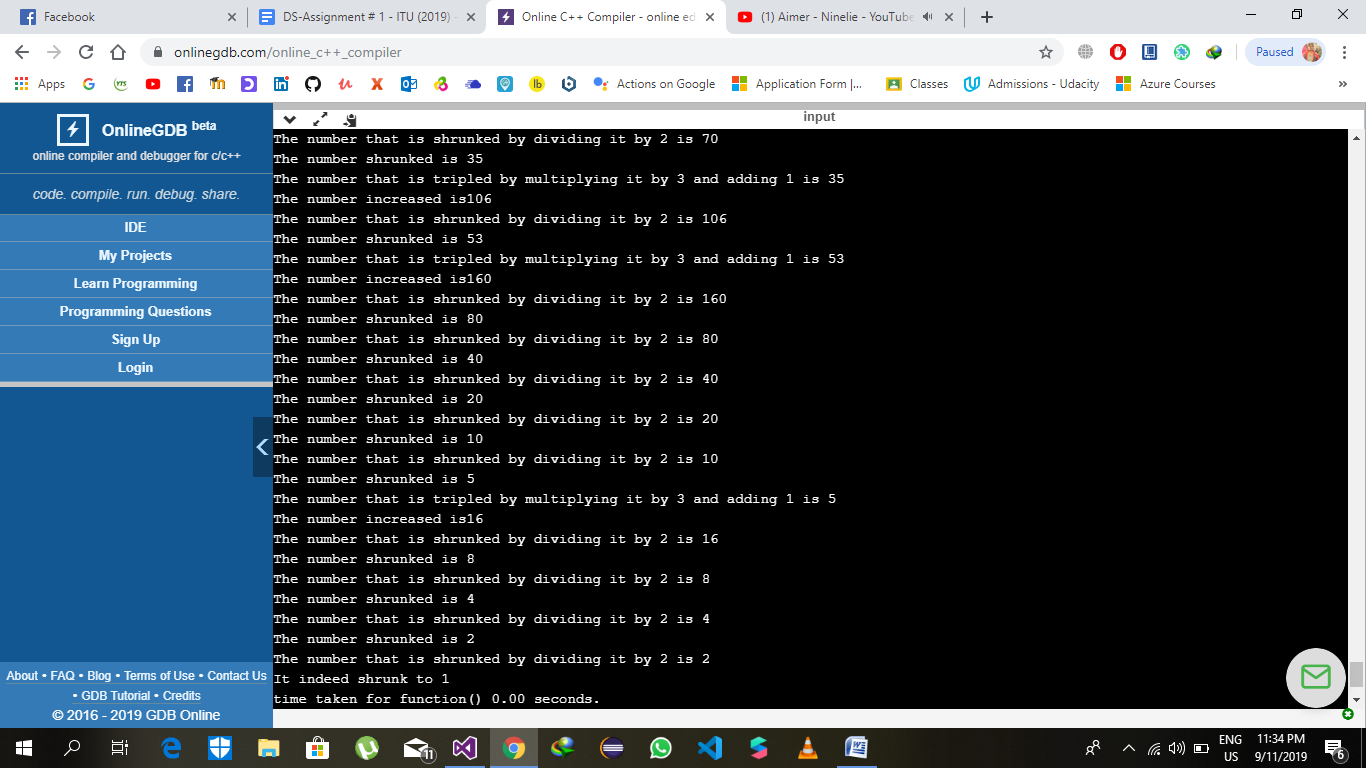




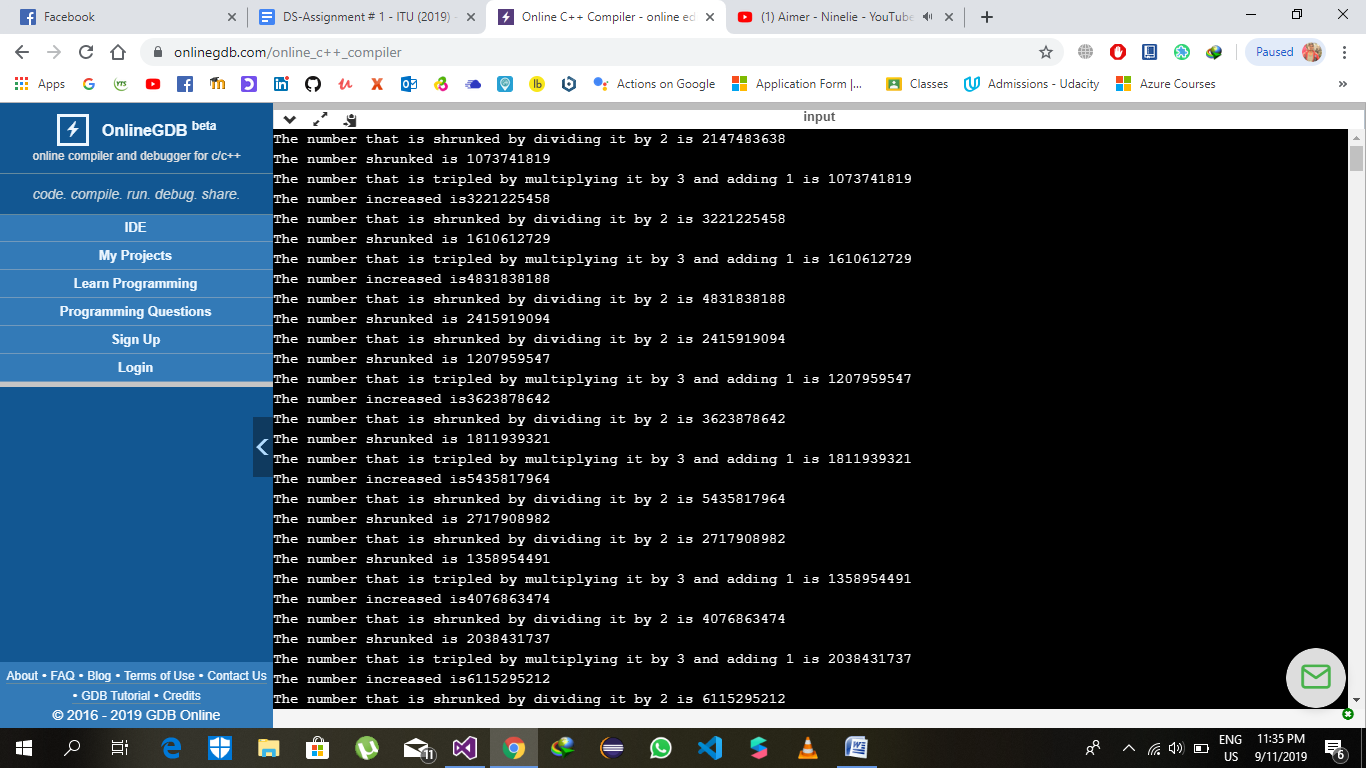
The number 9 = 2147483639

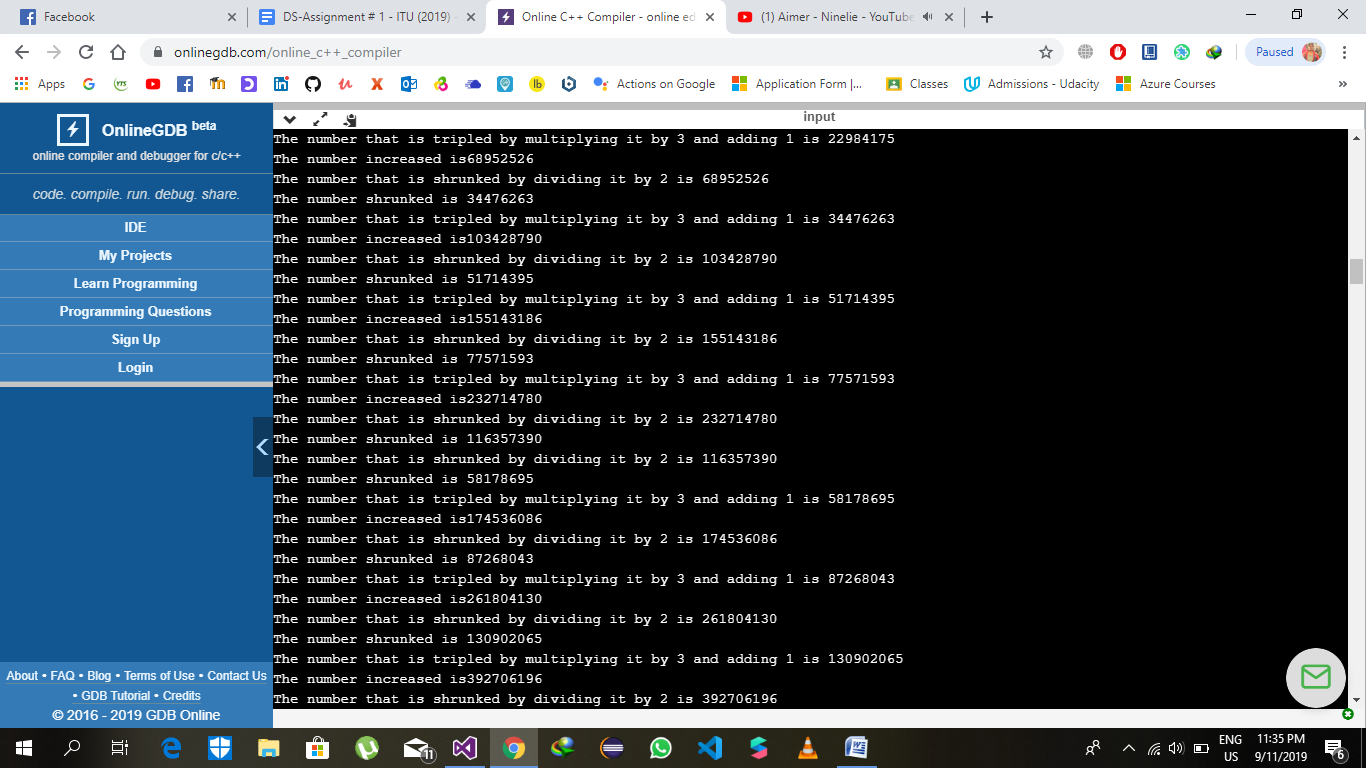


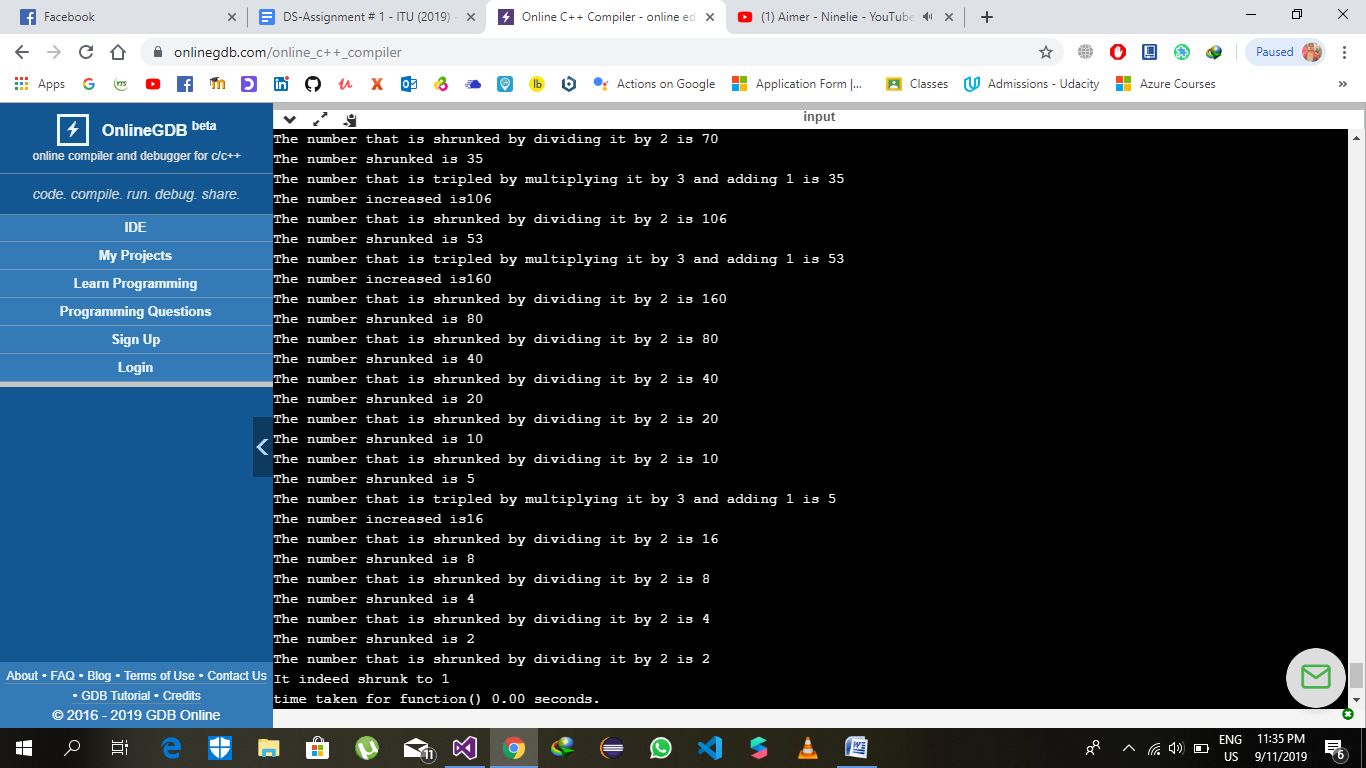




The number 10 = 2147483628







The program was to much time to check all the possibilities of integers. I kept the program running for 14 hours but still there was more time required to test all the possibilities. The program will take many light years to test all possibilities.

Part b)

Problem 4)

Ans : lets consider that A’s statement. He’s saying

that B is Knight. Lets consider that he’s telling the

truth. Then, B’s a Knight and this makes him a

KNAVE. So, it means he’s lying, which makes his

statement the “B is a Knight” false. So, both of them

are lying and are knaves. Now, to make it more

simple , let’s consider A’s statement false. Which

makes him a Knight and B’s a Knave …..BUT

KNIGHTS never lie. So, what A’s saying about B is

also a LIE. So, it proves that BOTH ARE LYING.

Problem 5)

a. Ans: The point C makes it clear that Treasure is

not in the Kitchen. So, If its not in the kitchen then

the point b is a hoax. THERE IS NO TREE IN

FRONT YARD WHICH’s AN ELM. The point D

“The tree in the front yard is an elm or the treasure is buried under the flagpole.

” , From our deduction to this point, The LATTER

part of point D makes sense as There is no ELM

TREE. Point E states an other “IF statement” BUT

WE DON’T NEED ANY. The Latter of part d is true.

And THE TREASURE IS BURIED UNDER THE

FLAGPOLE.

Problem 6:

a)

i) p 🡪 (~q v r)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P | Q | R | ~q | ~q v r | p🡪(~q v r) |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |

ii)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| p | Q | r | ~p | (p🡪q) | (~p🡪r) | (p🡪q) ∧(~p🡪r) |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 |

b)

i) The given claim states that if x is divisible by 150 then it is either a multiple of 30 or 90 (xor logic). This claim fails because if some number x is divisible by 150 then it can be a multiple of 30 and also 90. For example if x is 300 then it is a multiple of 90 and also of 30. So the above claim is not correct.

ii) The given claim states that if x\*y=0 then either x or y is equal to zero (xor). This claim fails because if x is zero and y is also zero then x\*y is also zero. Both x and y are zero not only one of them. So the above claim fails that if x\*y=0 then either x or y is zero (only one of them is zero). Because the condition in which both x and y are zero (not only one of them) satisfies x\*y=0 the above claim is incorrect.

iii) p= Islamabad is the capital of Pakistan

q=Faisal mosque is the largest mosque in the world

r=The capital of Pakistan has the world’s largest mosque

p ∧ q🡪r

p is true and q is false and there is a and operator between them so it ultimately evaluates to false. And the implication in which first statement is false then the second statement doesn’t matter and the whole implication becomes true.

Problem 7)

a)

He comes on time and in summer.

b)

Amina’s favorite food is not halwa puri or her most favorite drink is lassi.

c)

Jamil will not come to the class and he wakes up on time.

d)

x=3,y=0,z=10 and x\*y\*z!=0

e)

x\*y=0 or x! =0 and y! =0 or x=0 and y=0

Problem 8)

a)

~q🡪r

b)

p ∧ q ∧ r

c)

r🡪p

d)

p v q🡪r

e)

p ∧ q 🡪r

Problem 9)

a)

p= If they bat all 50 overs

q=Pakistan will win the match

p🡪q

Inverse:

If they don’t bat all 50 overs then Pakistan will not win the match

Converse:

q🡪p

If Pakistan will win the match then they bat all 50 overs

Contra-positive

~q🡪~p

If Pakistan will not win the match then they don’t bat all 50 overs.

b)

p=Winning the match against Bangladesh

q=Pakistan to qualify for semi-finals

p🡪q

Inverse:

~p🡪~q

If Pakistan do not win the match against Bangladesh, they will not qualify for semi-finals.

Converse:

q🡪p

If Pakistan qualifies for semi-finals then they will win the match against Bangladesh.

Contra-positive:

~q🡪~p

If Pakistan do not qualifies for semi-finals then they will not win the match against Bangladesh.

c)

p= To do all the assignments in DS

q=It is necessary to understand the material well.

p🡪q

Inverse:

~p🡪~q

It is not necessary to do all the assignment, to not understand the material.

Converse:

q🡪p

It is necessary to understand the material well then you have to do all the assignments in DS.

Contra-positive:

~q🡪~p

If it is not necessary to understand the material well , then you don’t have to do all the assignments in DS

Problem 10)

a)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p | q | p ∧ q | p nand q | ~(p nand q) |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |

b)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| p | q | ~p | ~q | p v q | ~p nand ~q |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |